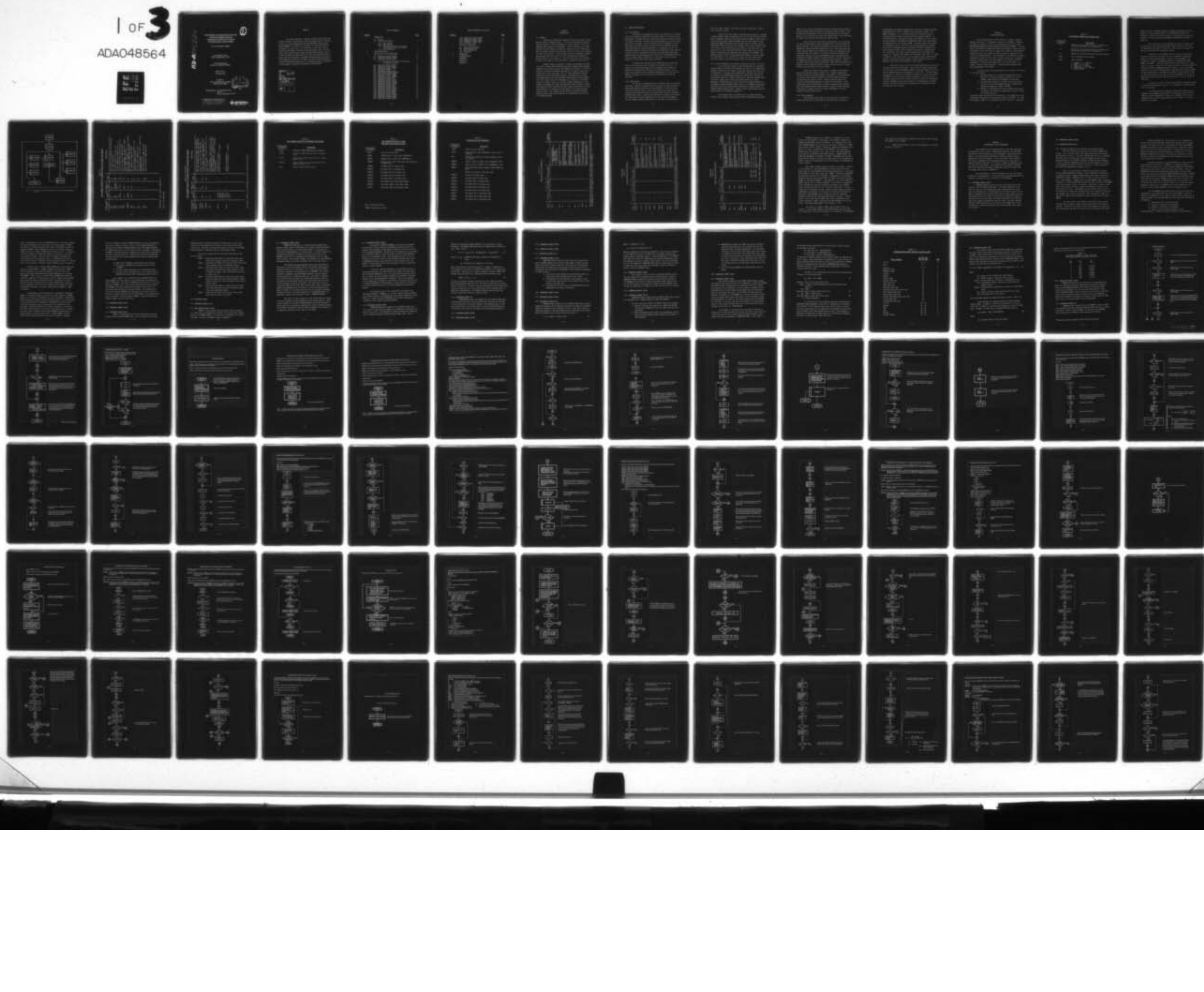
AD-A048 564 SPERRY RESEARCH CENTER SUDBURY MASS F/G 4/2
DEVELOPMENT OF CLOUD/FOG ANALYSIS AND APPLICATION SUBROUTINES F--ETC(U)
NOV 75 B R FOW, W D MOUNT DAAD07-74-C-0251
UNCLASSIFIED SCRC-CR-75-17 NL

1 OF 3
ADA048564



AD-A048564

(1)

DEVELOPMENT OF CLOUD/FOG ANALYSIS AND APPLICATION SUBROUTINES FOR EXPERIMENTAL PROTOTYPE AUTOMATIC METEOROLOGICAL SYSTEM (EPAMS)

B. R. Fow* and W. D. Mount*

Sperry Research Center
Sudbury, Massachusetts 01776

Final Technical Report
Contract No. DAAD07-74-C-0251

SCRC-CR-75-17

November 1975

Prepared for

WHITE SANDS MISSILE RANGE
NEW MEXICO 88002

*Present Address: Geo-Atmospherics Corp.
Box 177
Lincoln, Massachusetts 01773

SPERRY
RESEARCH CENTER

ABSTRACT

This report describes a computer software system called the Cloud/Fog Analysis system (CFAS), which was designed to be a subsystem of the U. S. Army's Experimental Prototype Automatic Meterological System (EPAMS). The function of the CFAS is to create and maintain information on cloud cover, fog and weather in near real-time on a mesoscale grid network covering a given geographical area. The data sources which the CFAS uses include teletype network transmissions of surface and upper air observations and cloud cover prognostications. State of the art techniques in automated meteorological data analysis were adapted and utilized in the CFAS. An overall system description as well as detailed descriptions of its component modules, principally via the medium of annotated flow diagrams, are presented.

TABLE OF CONTENTS

Section		Page
1	INTRODUCTION	1
	1.1 General	1
	1.2 Concept Philosophies	2
	1.2.1 Data Density	2
	1.2.2 Data Sources	2
	1.2.3 Data Interpretation and Analysis Techniques Utilized in the CFAS	3
	1.2.4 Design Approach	4
2	SYSTEM DESCRIPTION	6
3	DESCRIPTION OF CFAS SUBPROGRAMS	20
	3.1 Subprogram Element AFDINT	20
	3.2 Subprogram Element BAKUTM	21
	3.3 Subprogram Element BEGIN	21
	3.3.1 Storage and Retrieval Initialization Via Subroutine BEGIN	21
	3.4 Subprogram Element BLKIN	24
	3.5 Subprogram Element BLKOUT	24
	3.6 Subprogram Element CASES	24
	3.7 Subroutine CFEXEC	25
	3.8 Subprogram Element CFLAY	25
	3.9 Main Program Element CFMAIN	25
	3.10 Subprogram Element CFMAP	26
	3.11 Subprogram Element COMOBR	27
	3.12 Subprogram Element DEPCLD	27
	3.13 Subprogram Element FOG	28
	3.14 Subprogram Element GETOB1	28
	3.15 Subprogram Element GET1BW	28
	3.16 Subprogram Element GET1FW	29
	3.17 Subprogram Element ITMDIF	29
	3.18 Subprogram Element ITOJ	29
	3.19 Subprogram Element LAYCLD	29
	3.20 Subprogram Element MVLCOV	29
	3.21 Subprogram Element MOSECT	29
	3.22 Subprogram Element RAOB	29
	3.23 Subprogram Element RETOBR	30
	3.24 Subprogram Element SECTOR	30
	3.25 Subprogram Element SFDINT	30
	3.26 Subprogram Element STOREC	31

TABLE OF CONTENTS (Continued)

Section		Page
	3.27 Subprogram Element SYNOP	31
	3.28 Subprogram Element TOPS	34
	3.29 Subprogram Element UADINT	35
	3.30 Subprogram Element UTM	35
4	OPERATING INSTRUCTIONS	135
	4.1 Tasks and Data Inputs	135
	4.2 Mass Storage Files	135
	4.3 Core Requirements	139
	4.4 Sample Run	139
5	CONCLUSIONS	145
6	RECOMMENDATIONS	146
	REFERENCES	147
	APPENDIX I	I-1
	APPENDIX II	II-1

SECTION 1
INTRODUCTION

1.1 GENERAL

History has recorded many times that the turning point of major battles hinged on weather factors. In spite of technological advances, the effectiveness of men and equipment in today's armies is still very dependent upon vagaries of the weather. In fact, the need for accurate, detailed aviation weather to support air mobile forces has intensified. Most of the Army aircraft fly relatively slow and low over short distances for small time intervals. These aircraft are more vulnerable to weather because they cannot fly over it and lack sophisticated inflight and ground controls to fly through it. They are sensitive to air turbulence and most often require visual orientation to accomplish their mission.

Weather support information is either not adequately provided or involves a highly labor-extensive operation; however, major hardware components exist today and, given the appropriate software, it should be possible to develop an automatic meteorological system to satisfy the future needs of the field Army. The U. S. Army Atmospheric Sciences Laboratory at White Sands Missile Range is developing such a system called Experimental Prototype Automatic Meteorological System (EPAMS). This final report covers the development of a technology in automated analysis of surface, upper air, and pilot weather observations. Procedures that operate on a near real-time basis were developed to decode, interpret and analyze these data and then create and maintain cloud and fog information on a defined grid that applies at any specified geographical area. This computer program package is called a Cloud/Fog Analysis System (CFAS) and was written for compatibility with and use in the EPAMS program.

1.2 CONCEPT PHILOSOPHIES

1.2.1 Data Density

The quality and operational utility of the product of a meteorological data analysis system such as the CFAS varies as a function of the data density. The data density in turn varies extensively within the area of concern to the Army. In enemy controlled territory or so called silent area, almost no surface synoptic observations can be expected. On the other hand, the regions in which friendly troops are deployed are data rich, particularly in surface observations. Consequently, given 1) an automated system for collecting and processing these data in real time, 2) an assist from the information that could be obtained from satellites, weather radar and reconnaissance vehicles; and 3) an appropriate objective analysis system, useful information on cloud cover, fog, and weather conditions in the silent areas could be inferred.

At present the configurations of EPAMS does not contain provisions for a real-time input of satellite or weather radar data. To assure its continued operational utility throughout the evolution of available data sources and densities, the CFAS was devised in such a manner as to provide the best possible representation of cloud, fog and weather conditions under a wide range of observational sources and densities.

1.2.2 Data Sources

No one type of observational data completely depicts the Cloud/Fog fields. Optical, infrared and, most recently, microwave sensors on board meteorological satellites are most effective in providing total cloud cover and cloud tops. This, unfortunately, is not the case for multiple cloud bases and cloud or fog layers, which are often obscured beneath higher clouds. Analyses and predictions of global cloud coverage on a 25-mile grid has been automated at the Air Force Global Weather Central (AFGWC) at Omaha, Nebraska, and is available on a delayed time

basis for input to EPAMS. These data have been incorporated in CFAS as one of the data sources.

Cloud and fog data elements contained within regularly scheduled teletype weather network transmissions such as the hourly Aviation Weather Reports (AIRWAYS), Meteorological Aircraft Reports (METAR), three and six hourly Surface Synoptic Reports (SYNOP), and six hourly upper air wind radiosonde (RAOB) observations form the major part of the data sources. These data sources provide the most valuable near real-time weather observations used in the Cloud/Fog Analysis System. Also, provisions have been made for using unscheduled military surface, upper air, and pilot weather reports which are expected to be available to the EPAMS. The different types of data are processed and interpreted to extract Cloud/ Fog information in such a manner that allows for multi-level analysis from the surface to more than three kilometers above ground level.

1.2.3 Data Interpretation and Analysis Techniques Utilized in the CFAS

It is not at all unusual to have missing parts in a standard weather message transmission that degrades the value of an observational sequence. Further, not all pieces of available data have the same value or reliability. Procedures were established to determine the validity or ranking of the observational data and to maintain the integrity of the Cloud/Fog data base. Special observations, which are reported when a significant change occurs were given greater weight than routine messages. Consideration was given to the type of routine observations. For example, aviation weather reports given in the AIRWAYS sequence were given more value than meteorological observations contained within the synoptic weather sequence code. Likewise, direct observational data was given priority over inferred information.

Since weather data is perishable and its representativeness decreases with distance, older and more distant observations were ranked

lower in value. Also, those weather messages containing either errors or missing data were given a weighting that reduced their influence. These procedures were developed to rank the observational data in a relative manner so as to allow the most basic and reliable data to exert the greatest impact on the objective analysis of clouds and fog.

Both clouds and fog are highly discontinuous in the horizontal and vertical which greatly limits the application of standard objective analysis methods. The vertical distribution and layering of clouds was depicted by combining surface observations such as low, middle and high clouds, present weather and visibility with upper air observations from radiosondes and pilot reports. Although the vertical development and horizontal continuity of clouds are dependent upon cloud type and the interrelation between dynamical variables (winds and vertical velocities), no attempt was made to study and incorporate these features. Terrain was taken into account at each grid point in order that the analyzed cloud heights could be presented relative to above ground level as is usually desired by pilots flying via visual contact.

Within the scope of this effort, no attempt could be made to develop new objective analysis methods or to establish quantitative figures of merit for existing schemes. Much remains to be accomplished in a better understanding and formulation of more sophisticated and accurate objective analysis models. Special attention needs to be focused on technique performance for discontinuous variables in data-sparse areas and in devising ways to extract valuable weather information from nonmeteorological Army personnel to help close the data gaps. These types of studies are basic to future improvements in mission support involving clouds and fog.

1.2.4 Design Approach

The underlying philosophy of this effort was to proceed in a manner that got the job done within the time available by reducing,

interpreting, and analyzing weather data to provide a computer automated Cloud/Fog Analysis that is compatible with the present stage of the EPAMS development. It is recognized that vastly different levels of sophistication exist within the many individual program elements. To the extent possible, existing techniques were adapted for use including selected elements of the AFGWC 3-D Nephanalysis Model which was modified and adapted for use in the Army's EPAMS. Two major options are provided for analyzing the data. The first uses all available data to "create" an initial or subsequent analysis throughout the entire boundaries of a region called a window, which corresponds the region of responsibility of a field Army. The second uses the available data to "update" the analysis in a limited region within the field Army's area, which is called a subwindow. The overall logic of the Cloud/Fog Analysis System (CFAS), details of individual program elements, computer program listings, and operating instructions are documented in this report.

Many basic functions had to be accomplished before an analysis of cloud cover and fog could be made. Attention was directed towards insuring that each of these intermediate steps functioned properly so that the overall computer program system worked as a total unit. The CFAS program was designed in a modular or subroutine construction to allow for changes in EPAMS and to easily accept future improvements in program elements. The same attention and level of effort could not be given to each program element. Whenever a choice was necessary, increase emphasis was placed upon those functions that would be fundamental to any objective scheme for analyzing fog or clouds.

SECTION 2
SYSTEM DESCRIPTION

The Cloud/Fog Analysis System (CFAS) is a computer software package consisting of 30 subprograms coded in the language of FORTRAN V. The CFAS was designed to be one of the subsystems of the U. S. Army's Experimental Prototype Automatic Meteorological System (EPAMS). The function of the CFAS is to create and maintain information on cloud cover, fog and weather in near real-time on a square grid covering a user-specified geographical area from standard surface and upper air observations and cloud cover prognostications. The development of the CFAS was essentially engineering in nature in that state of the art technology in automated analyses of meteorological data was adapted to the specific needs and requirements of the principal application of the system (CFAS), i.e., support of Army aviation operations. These modified techniques were then incorporated in the system.

The data sources that were specifically considered in the design of the CFAS include:

1) Selected elements from scheduled teletype network transmissions of surface and upper air observations such as AIRWAYS, SYNOP, METAR, and RADIOSONDE code messages.
2) The three-hour prognosis of layered cloud cover produced by the Air Force Global Weather Central's (AFGWC) Three Dimensional Nephanalysis Model (3D-NEPH).
3) Elements of nonscheduled and special weather observations and reports with elements corresponding to those in either of the above sources.

These data inputs are referred to collectively in this report and in the programming documentation as observations-reports or OBS/REP. The input data elements common to all OBS/REP are given in Table 2-1. Input data

TABLE 2-1

DATA ELEMENTS COMMON TO ALL OBS/REP TYPES

Variable Name In CFAS	Description
IX	Distance in the eastward direction of OBS/REP site from window reference point (hectometers).
IY	Distance in the northward direction of OBS/REP site from window reference point (hectometers).
IZ	OBS/REP site elevation above mean sea level (meters).
ITIME	Time of OBS/REP (0-1439 minutes).
ITYPE	Type of OBS/REP 1 = AIRWAYS, -1 if a SPECIAL 2 = METAR, -2 if a SPECI 3 = SYNOP, -3 if a SPECIAL 4 = RADIOSONDE, -4 if a SPECIAL 5 = AFGWC, 3D-NEPH

elements specific to AIRWAYS, METAR, and SYNOP type OBS/REP are given in Table 2-2. Those elements specific to RADIOSONDE type OBS/REP are contained in Table 2-3, and those specific to the three-hour prognosis of layered cloud cover produced by the AFGWC'S 3D-NEPH model are given in Table 2-4.

From a collection of these OBS/REP, the CFAS will create a new or update an existing Cloud/Fog Data Base (CFDB) consisting of fifteen (15) parameters (described in Table 2-5) at each point on a square grid covering a user-specified geographical window.

In addition to the OBS/REP, the user must provide the items specified in Table 2-6 via either the list of arguments in SUBROUTINE CFEXEC, (the CFAS subprogram which interfaces with the EPAMS), DATA statements in CFEXEC, or PARAMETER statements in SUBROUTINES CFEXEC, COMOBR, and CFMAP. After more detailed testing and evaluation of CFAS, many of the items in Table 2-6 will become constant or move from subroutine arguments to DATA or PARAMETER statement elements. For the time being, however, and in view of the experimental nature of the host system, EPAMS, it seemed appropriate to incorporate the capability within CFAS of allowing the user to easily change the more critical parameters of the system. Suggested initial values for these items are given in the last column of Table 2-6.

The data and control flow among the principal subprogram elements of the CFAS are diagramed in Fig. 2-1. The following is a brief description of the system.

The CFAS receives task commands, operational parameters, and OBS/REP from the EPAMS through CFEXEC. The first task command that the CFAS must receive subsequent to startup of the EPAMS is to set up the OBS/REP files on the mass storage devices, (i.e., TASK=1). This job is carried out by BEGIN.

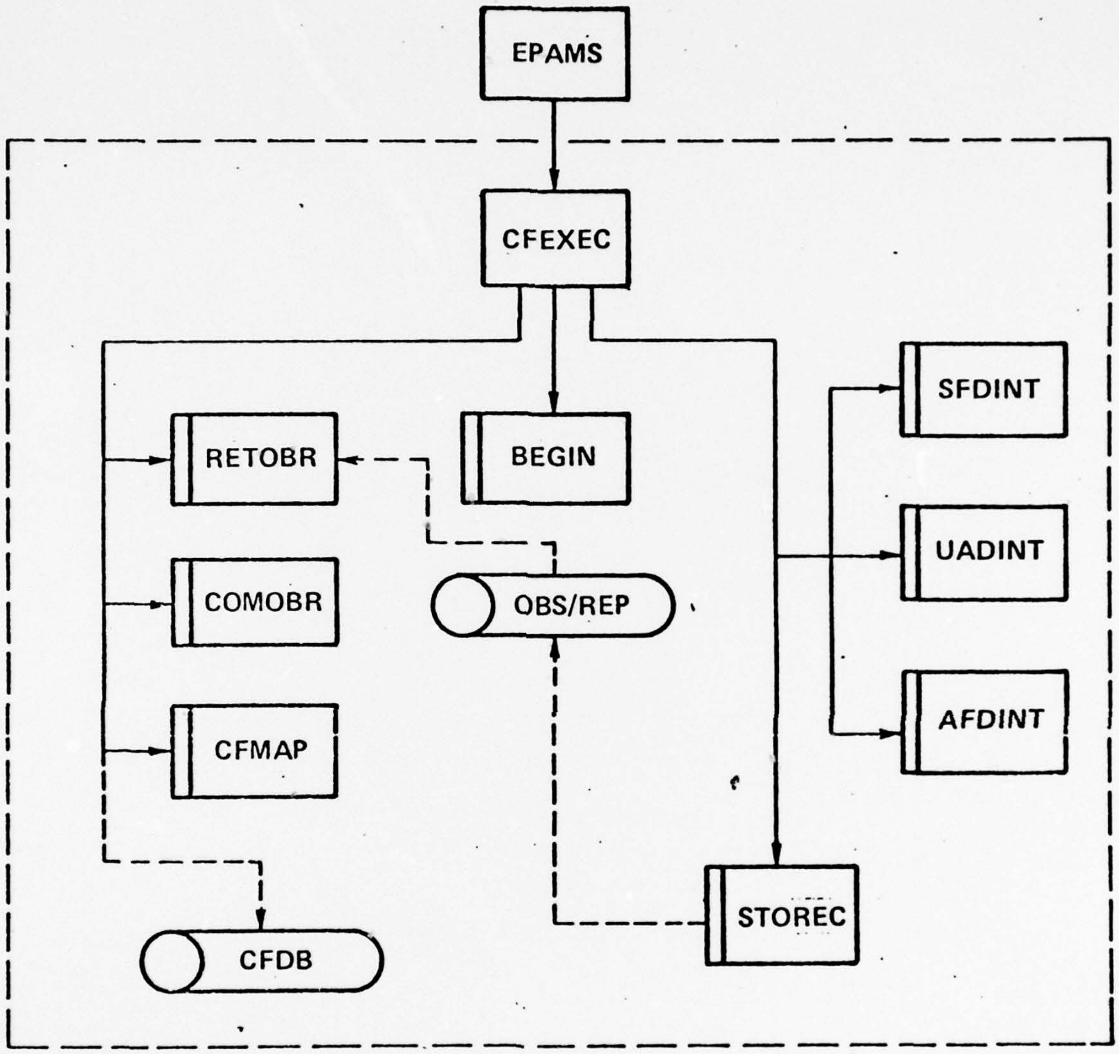


FIG. 2-1 Data and control flow among the principal subprograms in CFAS.

TABLE 2-2

DATA ELEMENTS SPECIFIC TO AIRWAYS, METAR, AND SYNOP TYPE OBS/REP

Variable Name In CFAS	Code Symbol Used AIRWAYS	METAR	SYNOP	DESCRIPTION
IDD	dd	ddd	dd	Wind direction 0-360 degrees from true north
IFF	ff	ff	ff	Wind speed, meters/sec.
IPPP	PPP	$P_H P_H P_H P_H$	PPP	Sea level pressure, millibars x 10
ITT	TTT	T'T'	TT	Surface temperature, degrees Kelvin x 10
ITD	$T_d T_d T_d$	$T'_d T'_d$	$T_d T_d$	Dew point temperature, degrees Kelvin x 10
ITSC			N	Total sky cover, 0-9 (WMO Code 2700)
IVIS	vvvv	VVVV	VV	Visibility (AIRWAYS - statute miles, METAR - meters, SYNOP - WMO Code 4377)
NWEA(7)	wwwwwww	w'w'	ww	Present weather (AIRWAYS - CFAS Code 1[1], METAR - WMO Code 4678, SYNOP - WMO Code 4677)
IPW	N.A.[2]	N.A.	W	Past weather, 0-9 (WMO Code 4500)
NH	"	"	N_h	Sky cover due to all low or middle clouds present, 0-9 (WMO Code 2700)
ICL	"	"	C_L	Low cloud type 0-9 (WMO Code 0513)
IH	"	"	h	Height above ground of the base of lowest cloud seen (WMO Code 1600)
ICM	"	"	C_M	Middle cloud type 0-9 (WMO Code 0515)
ICH	"	"	C_H	High cloud type 0-9 (WMO Code 0509)

[1] CFAS codes 1 and 2 given in Appendix II.

[2] N.A. = not applicable.

TABLE 2-2 (Continued)

DATA ELEMENTS SPECIFIC TO AIRWAYS, METAR, AND SYNOP TYPE OBS/REP

Variable Name In CFAS	Code Symbol Used AIRWAYS	METAR	SYNOP	DESCRIPTION
NS(10)	C_S	N_S	N_S	Sky cover due to cloud layer, 0-9 (AIRWAYS - CFAS Code 2, METAR and SYNOP - WMO Code 2700)
ICTS(10)		CC	C	Type of cloud layer, 0-9 (WMO Code 0500)
IHS(10)	hhh	h_sh_s	h_sh_s	Height of base of cloud layer (AIRWAY - 100's of feet, METAR and SYNOP - WMO Code 1677)
ITHN(10)	C_t	N.A.	N.A.	Cloud layer thickness indicator missing[3] if not thin, 1 if thin
ICLG	h_m	"	"	Ceiling designator - first two digits is the index number of the ceiling layer, third digit has following meaning: 1 = Measured, 2 = Aircraft, 3 = Balloon, 4 = Radar, 5 = Estimated, 6 = Indefinite.
ICLGV	"V" following hhh of ceiling layer	"	"	Characteristic of ceiling missing = not variable, 1 = variable
IVISC	"V" following vvvv	"	"	Visibility characteristic missing = not variable, 1 = variable

[3]The number -32768 is used throughout the CFAS to indicate missing data.

TABLE 2-3

DATA ELEMENTS SPECIFIC TO RADIOSONDE TYPE OBS/REP

Variable Name In CFAS	Description
IZ(30)	Altitude of RAOB reporting level, (meters).
IP(30)	Pressure at RAOB reporting level, (millibars x 10).
IT(30)	Temperature at RAOB reporting level, (degrees Kelvin x 10).
IDD(30)	Dewpoint depression at RAOB reporting level, (degrees Celsius x 10).
NRRL	Number of RAOB reporting levels.

TABLE 2-4

DATA ELEMENTS SPECIFIC TO AFGWC

THREE-HOUR CLOUD FORECAST FIELDS

Variable Name In CFAS	Description
NTCLC	Total sky cover 00-100.
MINBAS	Minimum base of clouds, AGL* (dekameters).
MAXTOP	Maximum top of clouds, AGL (dekameters).
LCOV(1)	Percent cloud cover in the layer from surface to 45 meters AGL.
LCOV(2)	45 meters AGL to 91 meters AGL.
LCOV(3)	91 meters AGL to 183 meters AGL.
LCOV(4)	183 meters AGL to 305 meters AGL.
LCOV(5)	305 meters AGL to 610 meters AGL.
LCOV(6)	610 meters AGL to 1067 meters AGL.
LCOV(7)	1067 meters AGL to 1524 meters AMSL**.
LCOV(8)	1524 meters AMSL to 1981 meters AMSL.
LCOV(9)	1981 meters AMSL to 3048 meters AMSL.

*AGL = above ground level.

**AMSL = above mean sea level.

TABLE 2-5

CLOUD/FOG DATA BASE PARAMETERS

Variable Name In CFAS	Description
NTCLC	Total sky cover (00-100).
NCEIL	Height ceiling layer, (dekameters, AGL), minus if variable.
NVV	Prevailing visibility at surface, (meters), minus if variable.
MINBAS	Height of base of lowest cloud, (dekameters, AGL).
MAXTOP	Height of top of highest cloud, (dekameters, AGL).
MSPWE	Most significant present weather element (WMO Code 4677).
	Percent cloud cover in the layer from
LCOV(1)	Surface to 45 meters AGL.
LCOV(2)	45 meters AGL to 91 meters AGL.
LCOV(3)	91 meters AGL to 183 meters AGL.
LCOV(4)	183 meters AGL to 305 meters AGL.
LCOV(5)	305 meters AGL to 610 meters AGL.
LCOV(6)	610 meters AGL to 1067 meters AGL.
LCOV(7)	1067 meters AGL to 1524 meters AGL.
LCOV(8)	1524 meters AGL to 1981 meters AGL.
LCOV(9)	1981 meters AGL to 3048 meters AGL.

TABLE 2-6

USER SPECIFIED CFAS PARAMETERS

Variable Name	INPUT METHOD: Argument List In CFEXEC	Data Statement Subprogram Name	Parameter Statement Subprogram Name	Description	Suggested Initial Value
TASK	X			Task requested by EPAMS (1-4) 1 = Set up OBS/REP storage files 2 = Input on OBS/REP 3 = Create a new CFDB 4 = Update the last CFDB	
TIME	X			Reference time of CFDB creation or update.	
XO	X			Eastward distance from the window reference point of the lower left hand corner of the subwindow to be updated, (km.).	
YO	X			Northward distance from the window reference point of the lower left hand corner of the subwindow to be updated, (km.).	
XLN	X			East-west length of the subwindow to be updated, (km.).	
YLN	X			North-south length of the subwindow to be updated, (km.).	
TYMOLD	X			Time of the oldest OBS/REP to be used in the creation or update (0-1439 min)	
DSP[1]	X			Maximum distance between OBS/REP to be combined into a best report (km.).	20.

[1]These parameters were made subroutine arguments soley for the convenience of testing and evaluating the analysis scheme. It is expected that tests and evaluations will show that these parameters are somewhat site specific and would, therefore, be more appropriately set via DATA statements as are the other site specific parameters.

(Continued)

TABLE 2-6

USER SPECIFIED CFAS PARAMETERS

Variable Name	INPUT METHOD Argument List In CFEXEC	Data Statement Subprogram Name	Parameter Statement Subprogram Name	Description	Suggested Initial Value
DIST(1)[1]	X			Distance constant used in analysis when convective clouds only are present (km.).	20.
DIST(2)[1]	X			Distance constant used in analysis when convective and middle clouds only are present (km.).	80.
DIST(3)[1]	X			Distance constant used in analysis for all other cases (km.).	100.
TYMC(1)[1]	X			Time constant used in analysis when convective clouds only are present (min.).	50.
TYMC(2)[1]	X			Time constant used in analysis when convective and middle clouds only are present (min.).	120.
TYMC(3)[1]	X			Time constant used in all other cases.	150.
ISSQ(5)[1]	X			Cut-off X or Y distances from a grid point of a best report used in determining a CFDB parameter at the grid point (grid point units).	1, 2, 3, 4
NSSQ[1]	X			Number of cut-off distances used.	4
NBKOUT	X			The block number in the CFDB file where the created or updated CFDB is to be filed.	

[1]These parameters were made subroutine arguments soley for the convenience of testing and evaluating the analysis scheme. It is expected that tests and evaluations will show that these parameters are somewhat site specific and would, therefore, be more appropriately set via DATA statements as are the other site specific parameters.

(Continued)

TABLE 2-6

USER SPECIFIED CFAS PARAMETERS

Variable Name	INPUT METHOD Argument List In CFEXEC	Data Statement Subprogram Name	Parameter Statement Subprogram Name	Description	Suggested Initial Value
IDENT(10)	X			Ten words of identification information that precedes the created or updated CFDB on the file.	
XREF		CFEXEC		East-west UTM grid coordinate of the lower left hand corner of the CFDB window (km.).	Site specific
YREF		CFEXEC		North-south UTM grid coordinate of the lower left hand corner of the CFDB window (km.).	Site specific
CMRD		CFEXEC		Central meridian of the window,degrees	Site specific
GRDPH(IJP)[2]		CFEXEC		Terrain elevation above mean sea level of the grid points.	Site specific
MNBR		CFEXEC		Minimum number of best reports within the largest cut-off distance required to update a CFDB parameter at the grid point.	1
GRD			CFEXEC, CFMAP	Grid point spacing (km.).	25.
LNTHX			CFEXEC, CFMAP	East-west length of CFDB window, (km.)	600.[3]
LNTHY			CFEXEC, CFMAP	North-south length of CFDB window, (km.).	600.[3]
NOBR			CFEXEC, COMOBR, CFMAP	Maximum number of OBS/REP to be used for a creation or update.	600.

[2]IJP = number of grid points.

[3]The value used for LNTHX and LNTHY in the checkout and debugging of CFAS was 200.

OBS/REP are input (i.e., TASK=2) to the CFAS one at a time through the array OBSRPT in the argument list of CFEXEC. OBSRPT is an integer one dimensional array of 143 words. The placement of the individual OBS/REP elements in this array is discussed in Section 4. CFEXEC selects either SFDINT, UADINT, or AFDINT to interpret the OBS/REP on the basis of its type. Interpretation of an OBS/REP is the calculation, determination or inference of one, some, or all of the CFDB parameters at the site of the OBS/REP from the data therein. AIRWAYS, METAR, and SYNOP type OBS/REP are interpreted by SFDINT. RADIOSONDE type data are interpreted by UADINT and cloud cover prognostications are interpreted by AFDINT. After the interpretation, the first 44 words of the OBS/REP are filed by STOREC.

With TASK=3, CFEXEC performs a creation of the CFDB. Creation is the generation of the CFDB parameters at every grid point in the window using all filed OBS/REP with associated observation or verification times (TIME) no older than a given time, TYMOLD. The qualifying OBS/REP are retrieved from the files by RETOBR. A determination of the appropriate distance and time constant (DIST and TYMC) to associate with each OBS/REP is obtained by RETOBR. A "best reports" list is then compiled by COMOBR from the list of qualified OBS/REP. A best report is determined at each OBS/REP site using the information contained in all OBS/REP within a specified distance of the site (DSP in Table 2-6). The techniques used in formulating the best reports provide a means for combining complementary information in or resolving conflicting interpretations among two or more OBS/REP close in space and time and, therefore, presumably depicting the identical meteorological situation. The best reports file is input data to CFMAP, which uses an exponential time-distance weighting scheme in analyzing the CFDB parameters at the grid points. The newly created CFDB is then output to a file.

The steps in an update, TASK=3, are the same as those in a creation. In the update, however, the CFDB parameters are calculated for a subsection of the window referred to as a subwindow. The parameters

which specify the subwindow are supplied by the user and input through the argument list of CFEXEC.

Detailed descriptions of each of the subprograms in the CFAS are given in Section 3.

SECTION 3

DESCRIPTION OF CFAS SUBPROGRAMS

The procedures and techniques embodied in the CFAS subprograms are described in this section. Annotated flow diagrams have been used extensively in the descriptions of the subprograms. Step identification labels in the flow diagrams correspond to symbolic statement labels, whenever the latter are present. In the cases where a step identification label is required in the flow diagram because of a page break or where the explicit depiction of a DO loop is needed, an alphabetic label is used in the flow diagram. It is hoped that this labeling correspondence will facilitate the reader's comparison of the flow diagram with the program code listings in Appendix I.

The major emphasis of the flow diagrams is to show the logical processes in the subprogram. To facilitate this, detailed descriptions of numerical calculations are not shown in the flow diagrams.

3.1 SUBPROGRAM ELEMENT AFDINT

SUBROUTINE AFDINT is used to process the layered cloud cover forecasts from the AFGWC 3D-NEPH model. The height boundaries of the first six layers in the 3D-NEPH model output are referenced to ground level and are identical to the boundaries of the first six layers in the CFDB. Consequently, the forecast cloud cover for these layers as well as the forecast total cloud cover, minimum base and maximum top require no processing for use by the CFAS. The height boundaries of the remaining layers are, however, referenced to mean sea level. This data together with the terrain elevation at the location of the 3D-NEPH data is used by AFDINT to calculate cloud cover in the seventh through ninth layers of the CFDB.

3.2 SUBPROGRAM ELEMENT BAKUTM

3.3 SUBPROGRAM ELEMENT BEGIN

3.3.1 Storage and Retrieval Initialization Via Subroutine BEGIN

Two files called File I and File J are used for OBS/REP data storage. File I contains recent data records while older data records are maintained in File J. To store an OBS/REP data record in the OBS/REP data base, the user simply calls subroutine STOREC and supplies the starting address of the OBS/REP. Subroutine STOREC stores the OBS/REP in File I, performs bookkeeping functions, and when necessary, transfers older OBS/REP data records from File I to File J in order to make room for more recent OBS/REP data records in File I.

File J is a ring buffer of NBLKFJ blocks, where each block contains NRPBFJ OBS/REP data records. These variables are initialized in subroutine BEGIN. Subroutine BEGIN defines the number of words per OBS/REP data record as NWDREC and thus the number of words per block in File J equals NWDREC * NRPBFJ which is called NWDBKJ. When it becomes necessary to transfer OBS/REP data records from File I to File J, the NRPBFJ oldest data records in File I are stored in table JBUF, in order of observation time, and the contents of JBUF are transferred to File J as the next block in the ring. After NBLKFJ blocks have been stored in File J, the next block is stored over the oldest and so on. Since all block transfers between core and mass storage File J are through buffer JBUF, the user should insure that the dimension of JBUF is greater than or equal to NWDBKJ.

The size of File J will depend on the amount of mass storage available and the number of hours of old data which the user wishes to maintain. The size of the blocks in File J will depend on the amount of core storage which the user can allocate for the dimensioned table JBUF.

Subroutine STOREC will be called upon to store OBS/REP data records having random observation times during the past several minutes or hours. The purpose of File I is to maintain the most recent NINTAB OBS/REP data records in terms of observation time. Since recent date is likely to be the most valuable data, File I is structured in a manner which facilitates usage of recent OBS/REP's. Blocks in File I contain OBS/REP's observed in sub-areas of the grid map. Retrieval of a block from File I brings all the OBS/REP's for that sub-area into core in one mass storage-to-core transfer.

As mentioned above, the oldest OBS/REP data records are transferred from File I to File J on a NRPBFJ at a time basis whenever a block in File I becomes full. Since subroutine STOREC can store OBS/REP data records only in File I, File I must be large enough to carry all OBS/REP data records having observation times during the past several minutes or hours plus an additional NRPBFJ OBS/REP data records. OBS/REP data records received by subroutine STOREC which have observation times older than the most recent OBS/REP in File J will not be stored in File I and thus not included in the OBS/REP data base. From the above, it can be seen that one of the functions of File I is to serve as a buffer between the random arrival of OBS/REP's for random observation times and File J which contains OBS/REP's sorted by observation time.

For each OBS/REP data record stored in File I, four words are maintained in a two-dimensional core array called ITABLE. The first dimension of ITABLE must be four, and the second must be greater than or equal to NINTAB. The four-word ITABLE entries contain the following for an OBS/REP:

1) observation time in minutes (0-1439),
2) relative X coordinate in hectometers,
3) relative Y coordinate in hectometers,
4) pointer to block and record number in File I.

Four-word entries in ITABLE are always maintained in sorted order in

terms of observation time with the OBS/REP having the most recent observation time represented by the four words in the first column of ITABLE. The contents of ITABLE are updated each time a new OBS/REP is stored via subroutine STOREC. NINI represents the number of four-word entries in ITABLE and thus the number of OBS/REP data records in File I. NINTAB and the second dimension of ITABLE are limited by the amount of core which can be used for ITABLE. This also limits the number of OBS/REP data records which can be maintained in File I. ITABLE permits the user to scan time and location of OBS/REP's before doing data transfers to retrieve them. Thus, efficiency is gained by making ITABLE as large as possible.

Blocks in File I contain NRPBFI OBS/REP data records per block, and hence contain NRPBFI * NWDREC = NWDBKI words per block. NRPBFI must not exceed 99. All blocks transferred to and from File I are through core buffer IBUF which must be dimensioned equal to or greater than NWDBKI. Blocks in File I contain recent OBS/REP data records for a specific sub-area of the grid called a sector. Within blocks, OBS/REP data records are sorted by observation time. This method of storage permits the user to retrieve all of the recent OBS/REP data records for a local geographic area in one mass storage-to-core transfer into IBUF. Consequently, efficiency is gained by making IBUF and NRPBFI as large as possible.

Establishment of sector boundaries is performed by subroutine SECTOR. In subroutine BEGIN, the size of the grid is defined by NROWS, NCOLS, and the number of hectometers per grid unit UTMPGD. Variable EDGE defines the minimum distance from outside grid points to the outer border of the outside sectors in hectometers. OBS/REP data records received from within this area are to be saved while those outside are to be discarded. Subroutine SECTOR uses the above information to divide the total area for which data is to be saved into square sectors. The maximum sector size is limited by the variable MAXGPS which specifies the maximum number of grid points per sector. Through MAXGPS, the user should select a sector size that is as large as possible. Blocks in

File I can contain a maximum of NRPBFI OBS/REP's for their corresponding map sector. Whenever a block in File I contains NRPBFI OBS/REP's, or whenever File I contains NINTAB OBS/REP's, the oldest NRPBFJ OBS/REP's are removed from File I to form a block in File J, and NINI is decremented by NRPBFJ. The above process is repeated until space becomes available in the File I block that was full. Large sector sizes have the advantage of reducing the number of mass storage-to-core transfers required when retrieving data.

The upper limit on MAXGPS is determined by the following:

1) The number of OBS/REP data records per block in File I (NRPBFI).
2) The rate of data observations in the data-rich areas.
3) The time interval for which File I is to maintain recent OBS/REP data records before transferring them to File J.

As an example, let us assume that each OBS/REP contains 44 words (NWDREC = 44) and that due to core limitations, the dimension of IBUF is 3750 words. The number of OBS/REP's per block in File I is thus determined (NRPBFI = 85). Let us also assume that we wish to maintain all OBS/REP's that were observed during the past hour in File I to facilitate rapid retrieval by time and location. Also assume that in the data-rich areas, OBS/REP's are generated at the rate of five per hour for a surface area the size of one grid square. The maximum sector size is thus 85/5 = 17 grid squares. MAXGPS = 17 provides subroutine SECTOR an upper limit for sector size.

3.4 SUBPROGRAM ELEMENT BLKIN

3.5 SUBPROGRAM ELEMENT BLKOUT

3.6 SUBPROGRAM ELEMENT CASES

CASES is a subprogram with six entry points which was taken directly from the AFGWC 3D-NEPH model. The six elements constituting

CASES are used to calculate the amount of cloud cover in two or three layers given the probabilities of clouds in the layers and the total cloud cover. Also taken into account when given is the low cloud amount and the presence of towering cumulus or cumulonimbus clouds.

The six entry points and the calculations performed by each are as follows:

- CASE1 - Calculates three layers of cloud cover given total cloud cover, assuming layers are completely random.
- CASE2 - Calculates two layers of cloud cover given total cloud cover, assuming layers are completely random.
- CASE3 - Calculates three layers of cloud cover given lowest cloud cover and total cloud cover, assuming a towering cumulus in layers 1 and 2 and random cloud cover in layers 2 and 3.
- CASE4 - Calculates three layers of cloud cover given lowest cloud cover and total cloud cover, assuming a cumulonimbus in layers 1, 2 and 3 and random cloud cover in layers 2 and 3.
- CASE5 - Calculates three layers of cloud cover given lowest cloud cover and total cloud cover, assuming layers 2 and 3 are completely random.
- CASE6 - Calculates two layers of cloud cover given lowest cloud cover and total cloud cover.

3.7 SUBROUTINE CFEXEC

3.8 SUBPROGRAM ELEMENT CFLAY

3.9 MAIN PROGRAM ELEMENT CFMAIN

CFMAIN is a main program that was used to drive the CFAS in the final stages of debugging and checkout. Since the CFAS is normally driven by the EPAMS, CFMAIN is not used and is, therefore, not a part of the CFAS. A listing of CFMAIN is given in Appendix I.

3.10 SUBPROGRAM ELEMENT CFMAP

SUBROUTINE CFMAP employs a time and distance exponentially weighted average value of observations lying within specified distances (i.e., cut-off distances or search square sizes) of a grid point for analyzing the CFDB parameters at the grid point. A further weighting of a particular observations influence on the grid point value is the OBS/REP's value which is incorporated multiplicatively with the exponential factor. This analysis concept is an adaptation of the techniques described by Mount, et al, Ref. 2; Barnes, Ref. 3; and Davis, Ref. 3.

The weighting technique employed herein gives recognition to differences in time and distance scales of the various meteorological conditions which may be encountered. One of these time and distance constants is selected in the weighting of each OBS/REP. The constants selected are dependent upon the probable presence of local cellular convective activity in combination with or in the absence of synoptic scale stratiform cloud systems. For identified convective clouds in the absence of both high and middle level stratiform clouds, the first time and distance constants are selected. For identified convective clouds and either but not both middle or high level clouds or identified showery type precipitation, the second set of time and distance parameters are used. In all other cases the third set of time and distance constants are used. The actual inspection of the OBS/REP wherein the applicable constants are determined is made in SUBROUTINE RETOBR.

The values of these constants as well as the number and values of cut-off distances or search squares, and the minimum number of observations required to analyze the CFDB parameters are system parameters whose values can be reasonably finalized only after an indepth evaluation of the CFAS. Consequently, they have been incorporated as variables to be supplied by the user in this version of the CFAS.

3.11 SUBPROGRAM ELEMENT COMOBR

The purpose of SUBROUTINE COMOBR is to form the best reports file from the list of time-qualified OBS/REP retrieved from the mass storage files by RETOBR. A best report is formed at the site of each of the time-qualified OBS/REP. Our definition or conception of a best report is a synthesized report in which the values for each of the CFDB parameters is a most probable value. The most probable value is the one selected from a list of values of that parameter obtained from a group of OBS/REP lying within given distance of the best report site. The details of the process by which the most probable value is selected are shown in the flow diagram.

A critically important parameter in the selection process is the maximum distance, i.e., DSP, from the best report site that within which candidate OBS/REP must lie. This value must be such that each of the candidates can, with reasonable assurance, be assumed to have been witness to the same meteorological situation. Time is also important in this regard, and its impact is accounted for in the fact that the list of candidate OBS/REP are time qualified. Second order time differences are also accounted for directly in the most probable value selection process.

Our ultimate intent in the incorporation into CFAS of the techniques embodied in COMOBR was 1) to have a logical means for selecting a correct value for a parameter when there existed conflicting information in a group of proximate OBS/REP, and 2) combine complementary information in the group of proximate OBS/REP.

3.12 SUBPROGRAM ELEMENT DEPCLD

SUBROUTINE DEPCLD was adapted from the AFGWC-3D NEPH model. It is used to convert dew point depression, temperature and pressure into percent cloud cover. The temperature, pressure and dew point spread are used to compute condensation pressure spread (CPS) values. CPS is

defined as the pressure change required for an air parcel to attain saturation. Uncorrected CPS values (Cu) are computed first according to Eq. 1, Refs. 1 and 5,

$$Cu = (T - T_d)_L \; [-4.9 - 0.93(P_L/1000) - 9.(P_L/1000)^2] \qquad (1)$$

where $(T - T_d)_L$ = Temperature-dew point spread at the midpoint of a layer.

P_L = Pressure at the midpoint of the layer.

Next, a multiplicative correction factor, K , based on temperature at the midpoint of the layer, is calculated and applied to Cu to obtain the correct CPS. Finally, Eq. 2 is used to compute an integer, INDEX, which provides the entry point into a CPS-cloud amount conversion table.

$$INDEX = 0.5 \; K \; Cu + 1.5. \qquad (2)$$

The CPS-cloud amount conversion tables are a set of empirical tables which were derived by Edson, Ref. 5. The tables are for 850 mb., 700 mb., 500 mb., and 300 mb. In order to obtain cloud amounts for each CFDB layer, values are taken from two of the above tables and the layered amount is obtained by interpolation. For midpoint pressure values greater than 850 mb., values from the 850 mb. table are used.

3.13 SUBPROGRAM ELEMENT FOG

SUBROUTINE FOG, which was adapted from the AFGWC-3D NEPH model, calculates sky cover due to fog from horizontal visibility and the type of fog as reported in the surface weather. The determinations of cloud cover amounts due to the various types of fog and the determinations of the height of the top of the fog layer utilize considerable empiricism.

3.14 SUBPROGRAM ELEMENT GETOB1

3.15 SUBPROGRAM ELEMENT GET1BW

3.16 SUBPROGRAM ELEMENT GET1FW

3.17 SUBPROGRAM ELEMENT ITMDIF

3.18 SUBPROGRAM ELEMENT ITOJ

3.19 SUBPROGRAM ELEMENT LAYCLD

SUBROUTINE LAYCLD constructs cloud layers from layered cloud data in AIRWAYS, METAR and the optional eight-group of SYNOP messages. LAYCLD will utilize assumed values for high, middle and low cloud base heights when the reported base heights are missing or are found to be inconsistent with other data in the OBS/REP. Other features incorporated in LAYCLD include:

1) a consistency check between the reported base height of a cloud layer and the reported genus of the cloud,
2) a determination of the KIND (i.e., high, middle or low) of cloud layer from the genus of the cloud as well as base height of the layer, and
3) a determination of the value of the OBS/REP based upon its completeness and internal consistency.

3.20 SUBPROGRAM ELEMENT MVLCOV

3.21 SUBPROGRAM ELEMENT NOSECT

3.22 SUBPROGRAM ELEMENT RAOB

SUBROUTINE RAOB, adapted from the AFGWC-3D NEPH model, analyzes temperature pressure and dew depression profiles from upper air soundings. Heights are computed for the significant levels in the report using the hydrostatic equation. Dew point depressions are calculated at each level where the reported value is missing according to Eq. 3.

$$T - T_d = .285\,(T - 273.) + 20.6, \qquad (3)$$

where T = temperature in ^{o}K,

T_d = dew point temperature in ^{o}K.

Pressure, temperature and dew point depression values are then calculated at the midpoints of each of the CFDB layers by linearly interpolating between adjacent radiosonde levels. A value determination of the OBS/REP based upon the fraction of temperatures and dew point depressions, which are reported as missing, is made in this routine. This value determination is then combined in SUBROUTINE UADINT with one based upon the number of CFDB layers for which cloud cover information could be inferred to arrive at the final OBS/REP value.

3.23 SUBPROGRAM ELEMENT RETOBR

This routine is used to retrieve interpreted OBS/REP from the file and inspect them to determine which time and distance constant in SUBROUTINE CFMAP is applicable to them and then tag them accordingly. The OBS/REP are retrieved in reverse chronological order starting with the one made closest to map time (i.e., TIME) and going backwards until the last one made prior to TYMOLD or the last one on the file is reached.

3.24 SUBPROGRAM ELEMENT SECTOR

3.25 SUBPROGRAM ELEMENT SFDINT

SUBROUTINE SFDINT, adapted from the AFGWC-3D NEPH model, directs the interpretation of AIRWAYS, METAR and SYNOP type OBS/REP.

The most significant features of SFDINT include the following:

1) Layered cloud data when reported in a SYNOP type OBS/REP override the information in the low, middle and high cloud data group.
2) Multiple present weather reports are accommodated, and the most significant weather element deduced and included as a CFDB parameter.

3) SUBROUTINES FOG, LAYCLD and SYNOP are called by SFDINT to construct cloud layers from fog, layered cloud and low, middle and high cloud data respectively. The information contained in each of these constructed cloud layers consists of the KIND of cloud layer (low, middle, high, fog, lowest or clear), the base and top of the layer and the percentage sky cover in the layer. These constructed cloud layers are then used to determine the percentage cloud cover in each of the CFDB layers, the minimum base and maximum top of the clouds.
4) Final and default OBS/REP value determinations are made herein.

3.26 SUBPROGRAM ELEMENT STOREC

3.27 SUBPROGRAM ELEMENT SYNOP

This routine, adapted from the AFGWC-3D NEPH model, analyzes the mandatory low, middle and high cloud information in SYNOP type OBS/REP. These data contain a limited amount of information from which layered cloud amounts can be determined. The only sources of layered data are the amount of all low (or middle) clouds present and the height above ground of the lowest cloud seen. If low, middle and high clouds are observed, the elements of the OBS/REP do not contain sufficient information to accurately define each cloud layer. Only the presence, absence or 50% probability of clouds in each height category can be determined from the data. In view of this, only an estimate of the most probable values of cloud cover, base and top of up to three cloud layers can be inferred from the data in a SYNOP OBS/REP.

The estimates of the base of the cloud layer and percent of coverage in the layer are made from the total sky cover, cloud type, low cloud cover, base of low or middle clouds and present weather using a complex decision tree embodied in SUBROUTINE SYNOP. In the course of

the decision process, probabilities of clouds within a height category are assigned as follows:

if cloud type = 0, 0% probability;

if cloud type $\neq$ 0, 100% probability;

if cloud type = missing, 50% probability.

In addition, if towering cumulus or cumulonimbus clouds are reported, this fact is noted. Having determined the probabilities of clouds within the height categories and given total sky cover and low cloud cover, towering cumulus or cumulonimbus when specified, SUBPROGRAM CASES employed to estimated coverage in layers assuming random distribution of the cloud elements within each layer.

The bases of the clouds within each of the three categories are computed in the following manner:

$$B_L = 2200 - 300 \times KWEA, \tag{4}$$

where B_L = base of low clouds (feet, AGL),

KWEA = a weather factor determined from present weather as per Table 3-1,

$$B_M = 10300, \tag{5}$$

where B_M = base of middle clouds (feet, AGL),

$$B_H = 35000 - 13000 \, (L/90), \tag{6}$$

where B_H = base of high clouds (feet, AGL),

L = latitude (degrees).

As is done in the case of layered cloud data interpretation in SUBROUTINE LAYCLD, whenever a cloud layer is determined to be overcast, cloud amounts for the remaining layers above are set equal to missing. Also, the cloud amounts for all layers from the surface up to and including the overcast layer are assigned values of zero. This procedure is necessary to insure that if these layers are not affected by subsequent decisions, they will reflect clear conditions.

TABLE 3-1

CONVERSION FROM PRESENT WEATHER TO WEATHER FACTOR

Type of Weather	WW (WMO CODE 4677 or 4678)	KWFA
———	0 - 9	0
Mist	10	1
———	11 - 14	0
Precip in sight	15	1
Precip in sight	16	2
Thunder	17	2
Squalls	18	2
Funnel clouds	19	3
Drizzle, past hour	20	1
Rain, past hour	21	1
Snow, past hour	22	1
Rain/snow, past hour	23	2
Freezing dirzzle/rain, past hour	24	1
Rain showers, past hour	25	2
Snow showers, past hour	26	2
Showers (hail/rain/snow), past hour	27	2
Ice fog, past hour	28	0
Thunderstorm, past hour	29	2
———	30 - 49	0
Drizzle	50 - 59	1
Rain	60 - 69	2
Snow	70 - 79	2
Showers	80 - 89	2
Thunder showers	90 - 99	3

3.28 SUBPROGRAM ELEMENT TOPS

This routine, adapted from the AFGWC-3D NEPH model, determines the tops of cloud layers whose constructions were begun in LAYCLD, FOG or SYNOP. Cloud thicknesses are first derived from the height of the base of the cloud layer, cloud amount in the base layer and the present weather factor KWEA described in Section 3.27 on SUBROUTINE SYNOP. Cloud tops are then obtained from the cloud thicknesses and the heights of the bases of the cloud layers according to Eq. 7,

$$T_L = B_T + D_T(K_F) + B_{SL}*S_T(K_F) + F_D*\{D_T(K_F + 1) + B_{SL}*S_T(K_F + 1)\}, \qquad (7)$$

where

T_L = height of top of cloud layer above terrain;

B_T = height of base of cloud layer above terrain;

$D_T(K_F)$ = cloud thickness, a function of cloud-weather index;

K_F = cloud-weather index, a function of cloud amount and weather factor;

$S_T(K_F)$ = cloud thickness coefficient, a function of cloud-weather index;

B_{SL} = height of base of cloud layer above mean sea level;

F_D = cloud amount (%) in base layer.

The relationship between K_F, $D_T(K_F)$ and $S_T(K_F)$ is given in Table 3-2.

Cloud tops are also computed exclusively from the weather factors (KWEA). If KWEA = 1, cloud tops are assigned to a value of 9,000 feet. If KWEA = 2, cloud tops are assigned to a value of 14,000 feet. If KWEA = 3, cloud tops are computed according to Eq. 8,

$$T_M = 40000 - 10000 * (\text{LATITUDE}/90) \qquad (8)$$

where

T_M = maximum height of cloud tops (MSL).

Finally, the maximum value of the two computed cloud tops is assigned as the value for the top of the cloud layer.

TABLE 3-2

RELATIONSHIP BETWEEN K_F, $D_T(K_F)$ AND $S_T(K_F)$

K_F	D_T	S_T
1	0	0
2	1287	0.13108
3	2843	0.25523
4	4323	0.41947
5	5864	0.62827
6	7636	0.87444
7	9843	1.11910

3.29 SUBPROGRAM ELEMENT UADINT

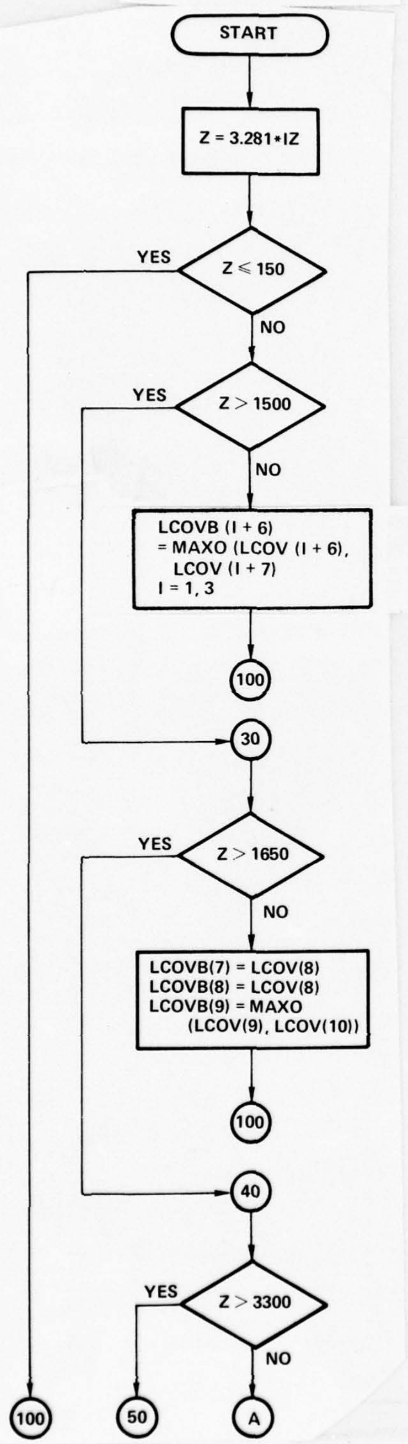
SUBROUTINE UADINT directs the analysis and interpretation of upper air soundings of pressure, temperature and dew point depression. Before calling SUBROUTINE RAOB, which analyzes the sounding, UADINT insures that the sounding is in the form that RAOB requires. After calling SUBROUTINE DEPCLD, which calculates the cloud cover in the CFDB layers from the analyzed sounding, the final value of the OBS/REP is determined in UADINT on the basis of the number of CFDB layers for which cloud cover or the absence thereof could be determined.

3.30 SUBPROGRAM ELEMENT UTM

SUBROUTINE UTM, obtained from the ASL-WSMR*, converts latitude and longitude to universal transverse mercator (UTM) easting and northing coordinates. UTM is called by BAKUTM, also obtained from ASL-WSMR, which calculates UTM coordinates from latitude and longitude.

*Atmospheric Science Laboratory, White Sands Missile Range.

SUBROUTINE AFDINT



Convert terrain height of 3D-NEPH data point to feet.

3D-NEPH data point terrain height less than 150 feet AMSL*.

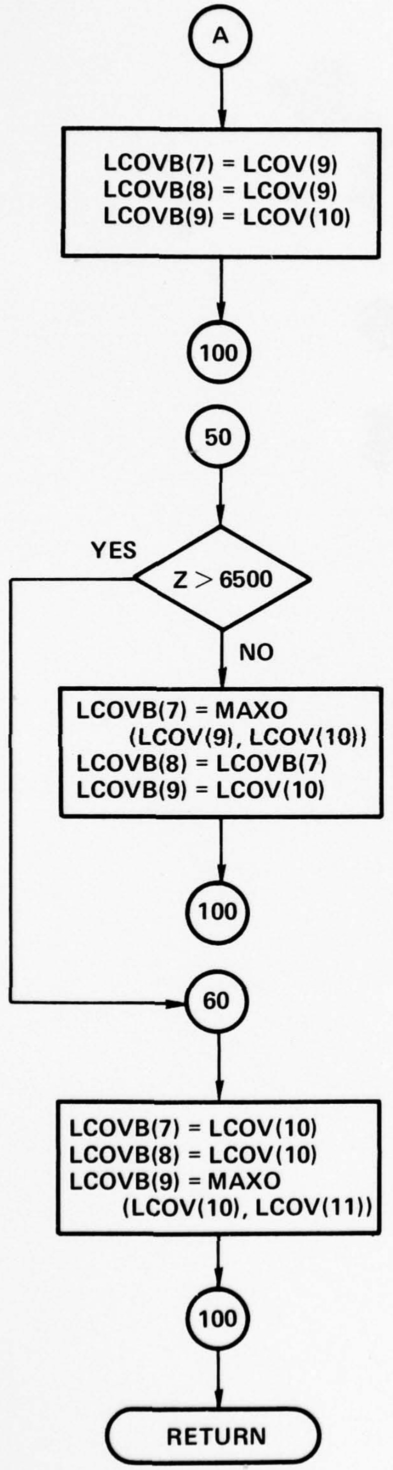
3D-NEPH data point terrain height greater than 1500 feet AMSL.

Set cloud cover in the seventh through ninth CFDB layers equal to the larger of the 3D-NEPH cloud covers in the corresponding or next higher level.

3D-NEPH data point terrain height greater than 1650 feet AMSL.

Set cloud cover of the seventh and eighth CFDB layers equal to cloud cover of the eighth 3D-NEPH layer and the cover in the ninth CFDB layer equal to the larger of the cloud covers in the ninth or tenth 3D-NEPH layer.

3D-NEPH data point terrain height greater than 3300 feet AMSL.



Set cloud covers in the seventh, eighth and ninth CFDB layers equal respectively to the cloud cover in the ninth, ninth and tenth 3D-NEPH layers.

3D-NEPH data point terrain height greater than 6500 feet AMSL.

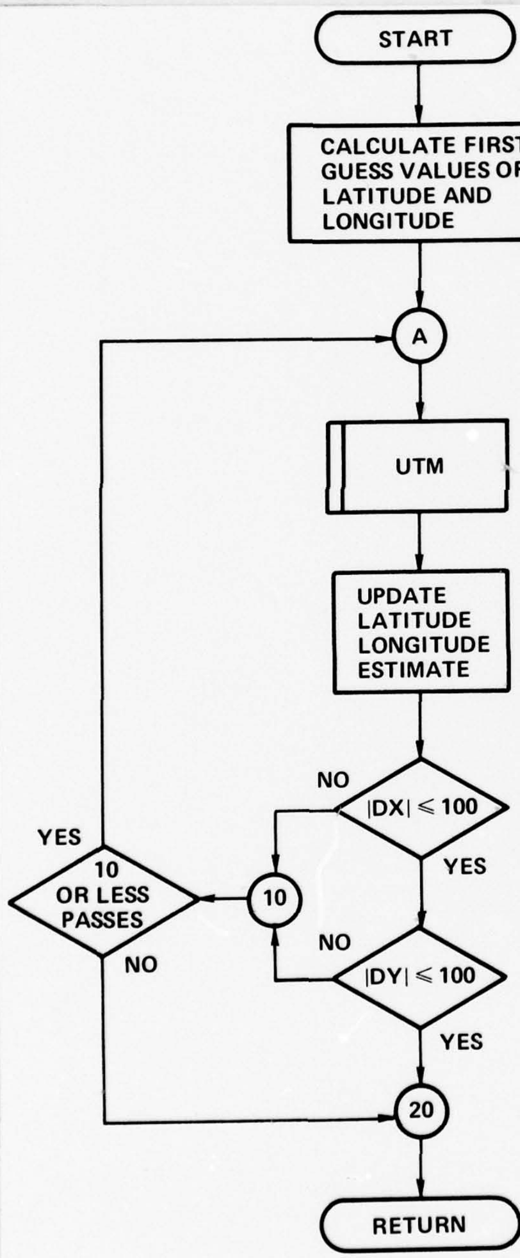
Set cloud cover in the seventh and eight CFDB layers equal to the larger of the cloud covers in the ninth and tenth 3D-NEPH layers and the cloud cover in the ninth CFDB layer equal to the cloud cover in the 3D-NEPH layer.

Set cloud cover in the seventh and eight CFDB layers equal to the cloud cover in the tenth 3D-NEPH layer and the cloud cover in the ninth CFDB layer equal to the larger of the cloud covers in the tenth and eleventh 3D-NEPH layers.

*AMSL = Above mean sea level.

SUBROUTINE BAKUTM (W, Z, X, Y, CMRD)

Inverse of UTM – Converts hundreds of kilometers to degrees.
A – Conversion factor (100's of km/radian along great circle)
RAD – Conversion factor (radian/degree)
CMRD – Central meridian in degrees
DWN, DZN, W, WN, Z, ZN – In degrees
DX, DY, X, XN, Y, YN – In 100's of km



Calculate UTM coordinates from estimated latitude and longitude.

Update latitude and longitude estimate from difference between calculated and given UTM coordinates.

Final value of latitude and longitude if absolute value of difference between calculated and given UTM X and Y coordinates both less than 100 meters or if this is the tenth correction.

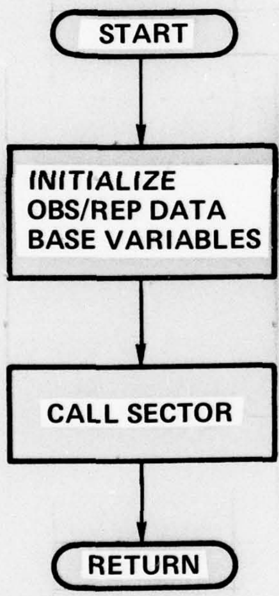
SUBROUTINE BEGIN

Begin initializes variables used by the routines which store and retrieve OBS/REP's in the OBS/REP data base.

NOTE – Unless otherwise noted – all distance measurements, UTM units, and UTM coordinates are carried in hectometers where 1 hectometer equals 100 meters.

NOTE – Unless otherwise noted – all times will be carried in minutes for a 1440 minute clock.

XREF and YREF must be in kilometers and must be supplied by the calling program.



Before calling BEGIN, the absolute coordinates of the lower left hand corner of the grid map must be initialized. The variables are XREF and YREF and are located in COMMON /MAP/.

See comments in BEGIN.

Establish the sector map corresponding to the blocks in file I.

SUBROUTINE BLKIN (NWDBLK, ISTART, NBKIN, LSFILE, ISTAT)

BLKIN transfers to core a block from a random access file that contains blocks that are all of the same size.

NWDBLK = No. of words per block in the file and the No. of words to be transferred to core on this call.

ISTART = Starting address in core where the block is to be transferred to.

NBKIN = No. of this block in the file. NBKIN = 1 is the first block No. in the file.

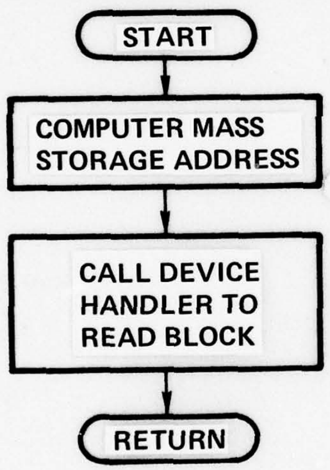
LSFILE = Logical system file No. (0-15).

ISTAT = Status returned to user. ISTAT = 0 indicates no errors. ISTAT = 1 indicates an error of some kind.

1108 disk version

Restrictions on this version of BLKIN

The status ISTAT returned to the user will always be zero since the FSTRD routine does not return any status information. FSTRD has its own error messages.



Return status to calling program.

Note — All CFAS mass storage to core transfers are through subroutine BLKIN. To implement CFAS on another computer, a new version of BLKIN having the above calling sequence will be required.

SUBROUTINE BLKOUT (NWDBLK, ISTART, NBKOUT, LSFILE, ISTAT)

BLKOUT transfers a block from core to a random access file which contains blocks that are all of the same size.

NWDBLK = No. of words per block in the file and the No. of words to be transferred from core on this call.

ISTART = Starting address in core where the block is to be transferred from.

NBKOUT = No. of this block in the file. NBKOUT = 1 is the first block No. in the file.

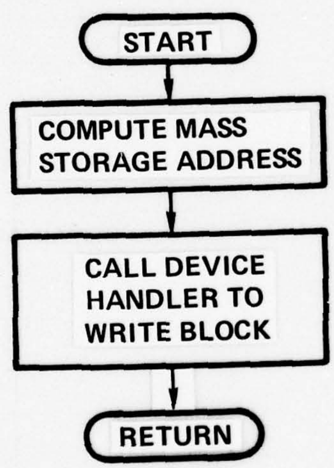
LSFILE = Logical system file No. (0-15).

ISTAT = Status returned to user. ISTAT = 0 indicates no errors. ISTAT = 1 indicates an error of some kind.

1108 disk version

Restrictions on this version of BLKOUT

The status ISTAT returned to the user will always be zero since the FSTWT routine does not return any status information. FSTWT has its own error messages.



Return status to calling program.

Note — All CFAS core to mass storage transfers are through subroutine BLKOUT. To implement CFAS on another computer, a new version of BLKOUT having the above calling sequence will be required.

SUBROUTINE CFEXEC (TASK, TIME, OBSRPT, XO, YO, XLN, YLN, LAST, TYMOLD, DSP, *DIST, TYMC, ISSQ, NSSQ, NBKOUT, IDENT)

This routine is the interface between the experimental prototype automatic meteorological system (EPAMS) and the cloud-fog analysis system (CFAS). In addition CFEXEC directs the interpretation of the surface and upper air observations and reports (OBS/REP) and the creation or updates of the cloud fog data base (CFDB).

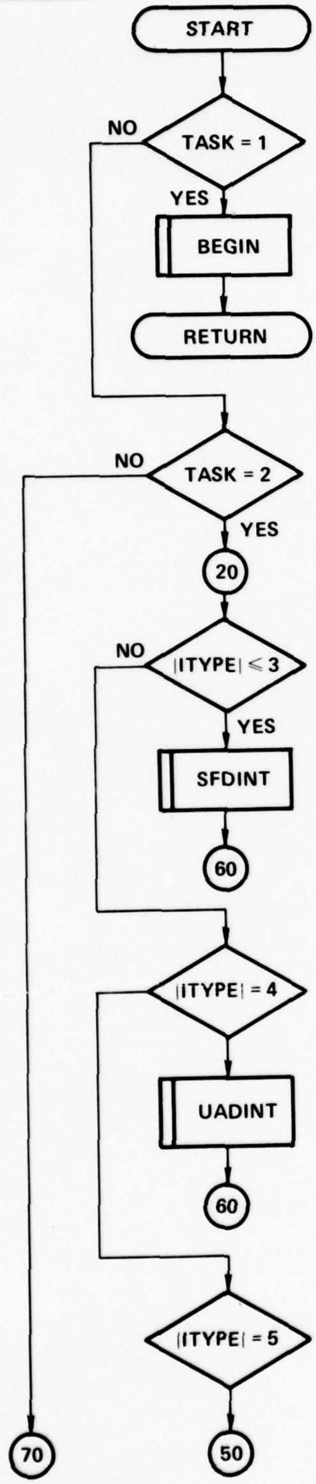
Input data (formal parameters)

TASK = Task requested by EPAMS
- 1 = Set up the OBS/REP storage files
- 2 = Input OBS/REP
- 3 = Create a new CFDB
- 4 = Update the latest CFDB on file

TIME = Reference time of CFDB creation or update
OBSRPT = OBS/REP
XO = Distance east from XREF of the lower left hand corner of the sub-window in the CFDB to be updated, km.
YO = Distance north from YREF of the lower left hand corner of the sub-window in the CFDB to be updated, km.
XLN = East-west length of updated sub-window, km.
YLN = North-south length of updated sub-window, km.
LAST = Sequence number of the last OBS/REP stored.
TYMOLD = Time of oldest OBS/REP to be used in a creation or update.
DSP = Maximum distance between OBS/REP to be combined into a best report, km.
DIST = Distance constants in weighting function, km.
- DIST(1) used when convective clouds only present.
- DIST(2) used when convective and middle clouds only are present or when showery type precipitation present or past weather.
- DIST(3) used for all other cases.

TYMC = Time constants in weighting function, minutes.
- TYMC(1) used when convective clouds only present.
- TYMC(2) used when convective and middle clouds only are present or when showery type precipitation present or past weather.
- TYMC(3) used for all other cases.

ISSQ = Search square sizes, no. of grid points.
NSSQ = No. of search squares used in analysis.
NBKOUT = Block no. in the CFDB file to which the creation or update is to be transferred.
IDENT = Ten words of user supplied identification information that precedes the cloud-fog-weather data on the file.

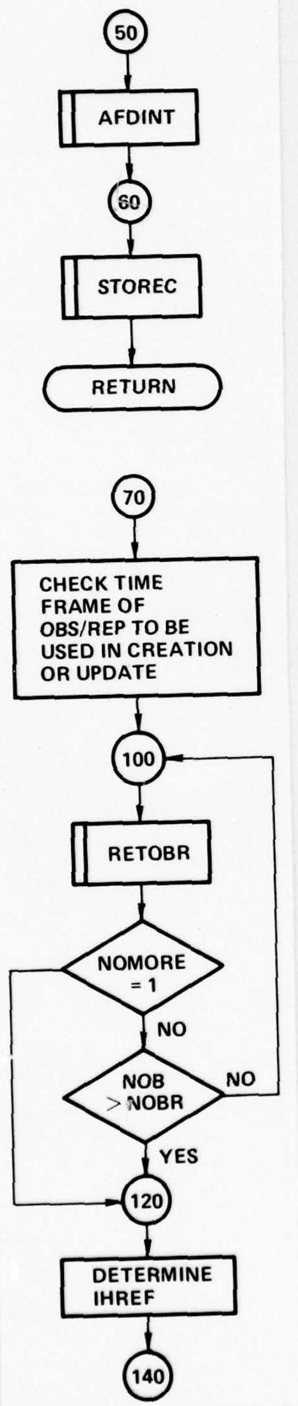


Initialize and set up OBS/REP files.

Come here to interpret OBS/REP.

Interpret surface type *OBS/REP*, *i.e.*, *AIRWAYS*, METAR, and SYNOP coded messages.

Interpret upper air type OBS/REP, i.e., RADIOSONDE coded messages.



Process forecast layered cloud coverage from AFGWC 3D-NEPH model.

File interpreted OBS/REP.

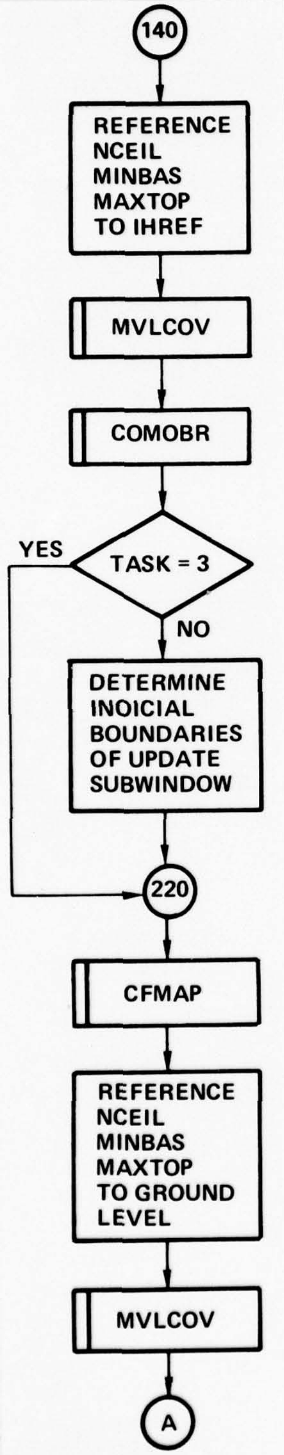
Assure that time of oldest OBS/REP (TYMOLD) to be used in creation or update is not more than 12 hours old.

Retrieve OBS/REP in reverse chronological order from present time (TIME) to TYMOLD. Also identify time and distance scale factors to associate with OBS/REP and tag accordingly.

Test for the presence of more OBS/REP within the allowed time frame on the file. Jump to 120 if there are no more.

Retrieve not more than NOBR OBS/REP.

Set reference altitude IHREF equal to the lowest of the altitudes specified in the list of OBS/REP or in the grid.



Reference ceiling, minimum base and maximum top of cloud layer to reference altitude IHREF.

Calculate cloud cover in layers referenced to IHREF from cloud cover in layers referenced to ground level at OBS/REP site.

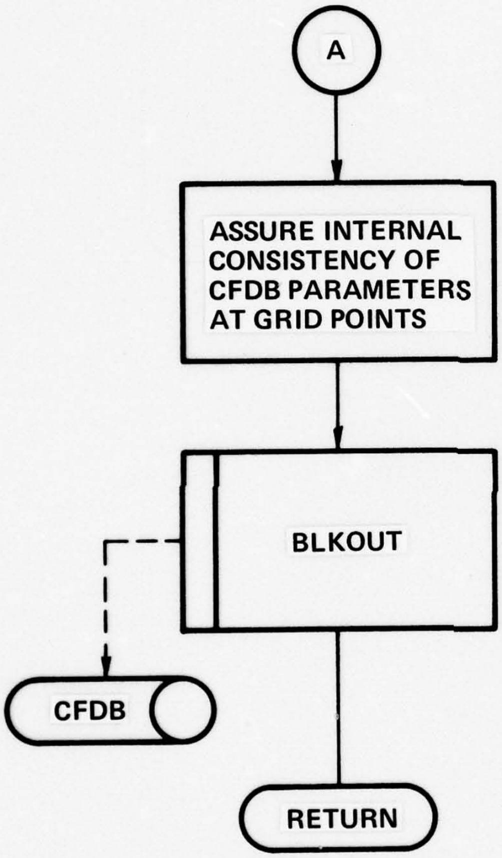
Form the best reports file from the list of qualified OBS/REP.

Convert user specified sub-window bounds and dimensions to indices of bounding grid points for an update. Use minimum and maximum values of indices for a creation.

Calculate CFDB parameters at the grid points lying within the indices of the bounding grid points.

Reference ceiling, minimum base and maximum top of cloud layer to ground level at each grid point.

Calculate cloud cover in layers referenced to ground level from cloud cover in layers referenced to altitude IHREF.



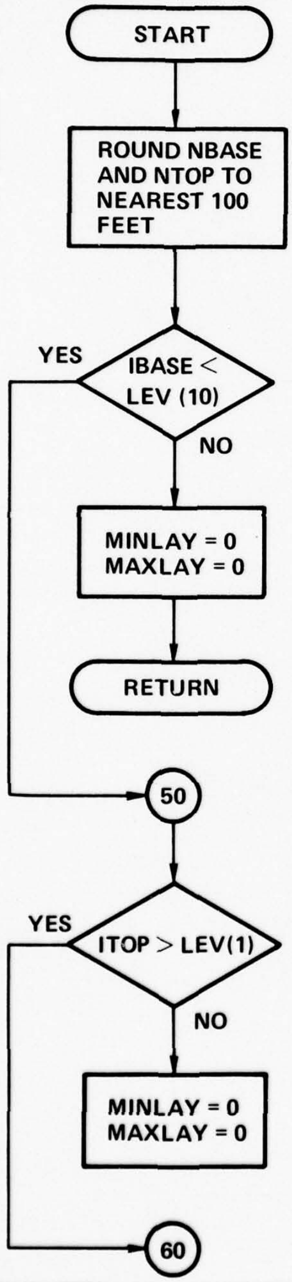
Assure that minimum base of clouds is less than maximum top of clouds and that total sky cover is equal to or greater than the largest amount of cloud cover in any layer.

Output the created or updated CFDB to user specified *file*.

SUBROUTINE CFLAY (NBASE, NTOP, MINLAY, MAXLAY)

Routine to find minimum and maximum CFDB layers influenced by cloud layers constructed from OBS/REP. 0 is returned if no CFDB layers are affected.

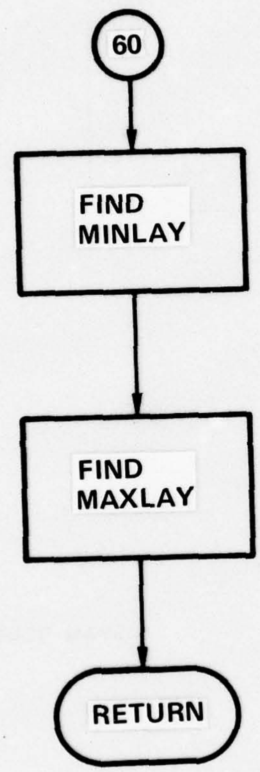
NBASE = Base in feet above terrain.
NTOP = Top in feet above terrain.
MINLAY = Minimum layer above terrain.
MAXLAY = Maximum layer above terrain.



IBASE and ITOP are set equal respectively to NBASE and NTOP rounded to the nearest 100 feet.

Return 0 for MINLAY and MAXLAY if the base of the cloud layer is higher than the top of the highest CFDB layer.

Return 0 for MINLAY and MAXTOP if the top of the cloud layer is lower than the base of the lowest CFDB layer.



MINLAY is equal to the index number of the CFDB layer whose top is higher and whose base is lower than IBASE.

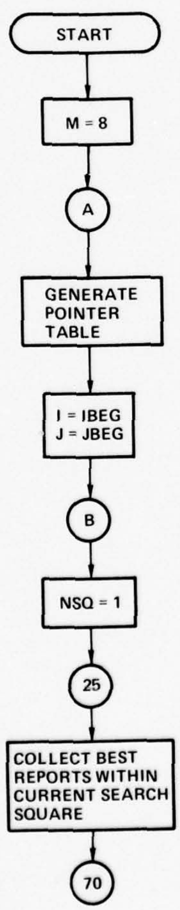
MAXLAY is equal to the index number of the CFDB layer whose top is higher and whose base is lower than ITOP.

SUBROUTINE CFMAP (IBEG, IEND, JBEG, JEND, DIST, TYMC, ISSQ, NSSQ, MNBR, *MTIME, NOB)

This routine uses the best reports generated by COMOBR to determine the CFDB parameters at specified grid points in the window.

Input data

IBEG = I index of left hand edge of window or sub-window.
IEND = I index of right hand edge of window or sub-window.
JBEG = J index of bottom edge of window or sub-window.
JEND = J index of top edge of window or sub-window.
DIST = Distance constants in weighting function, km.
TYMC = Time constants in weighting function, minutes.
ISSQ = Search square sizes, no. of grid points.
NSSQ = Number of search squares.
MNBR = Minimum number of best reports required to calculate CFDB parameters at a grid point.
MTIME = Map time (0 - 1440).
NOB = Number of OBS/REP.



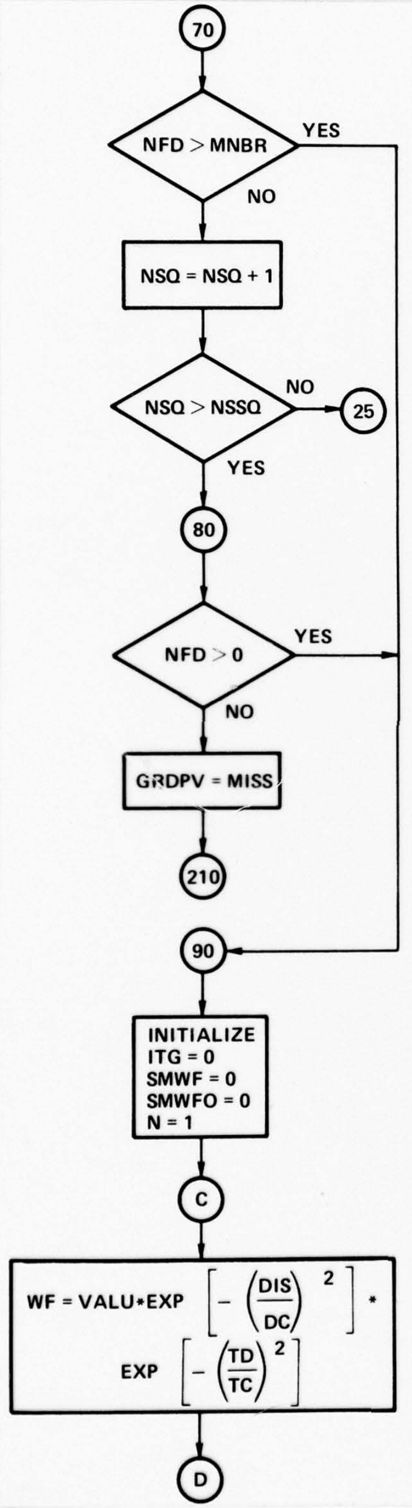
Initialize CFDB parameter index.

Search the best reports file and generate a pointer table to the best reports with a non missing entry for the current CFDB parameter.

Initialize grid point indices.

Initialize search square index.

Step through the pointer table and collect the best reports which lie within a square box of side length ISSQ (NSQ), called the search square.



Jump to 90 if the minimum number of best reports needed to analyze the current CFDB parameter at the grid point have been collected.

Increment the search square index.

Jump back to 25 and use the next larger search square if the largest one has not been used.

Jump to 90 if the number of best reports collected was at least 1.

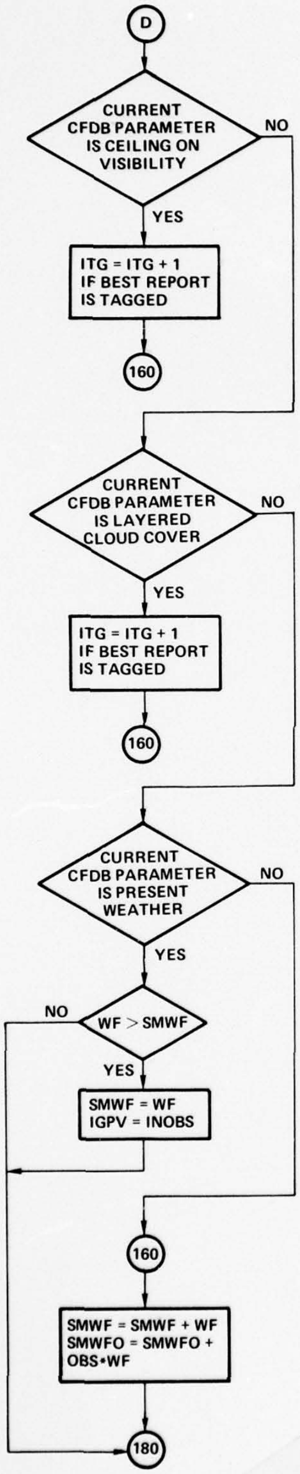
Set the current grid *point* value of the current CFDB parameter equal to missing, i.e., MISS = −32768 *if* no best reports were collected.

Calculate the weight factors, WF, corresponding to each of the best reports collected.

$$WF = VALU*EXP\left[-\left(\frac{DIS}{DC}\right)^2 - \left(\frac{TD}{TC}\right)^2\right]$$

Where

- VALU = value of the best report
- DIS = distance of best report site from the grid point
- DC = distance constant applicable to best report
- TD = time difference between time of best report and map time
- TC = time constant applicable to best report



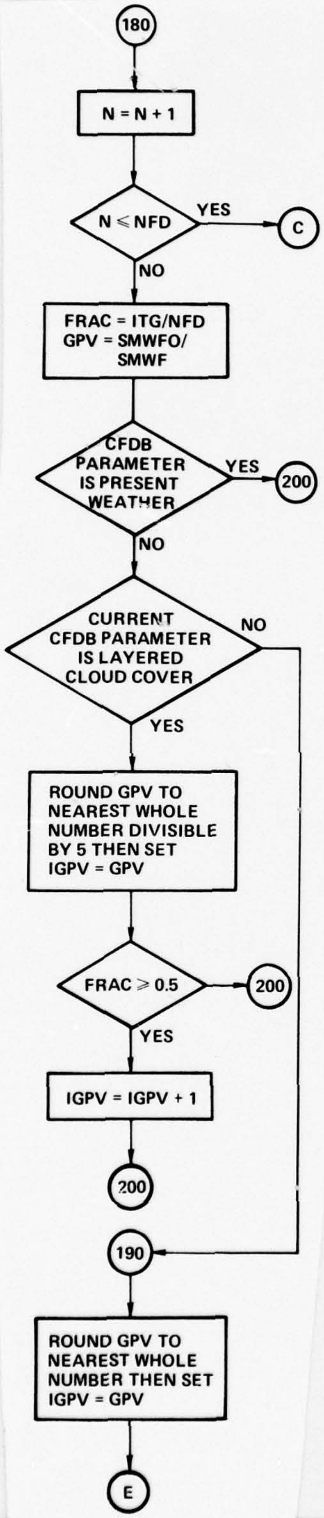
Increment tag count if the best report values for ceiling or visibility were tagged.

Increment tag count if the best report value for layered cloud cover was tagged.

For present weather only SMWF is the current largest value of WF.

Set grid point value for present weather equal to current best report value of present weather and SMWF to the current value of WF.

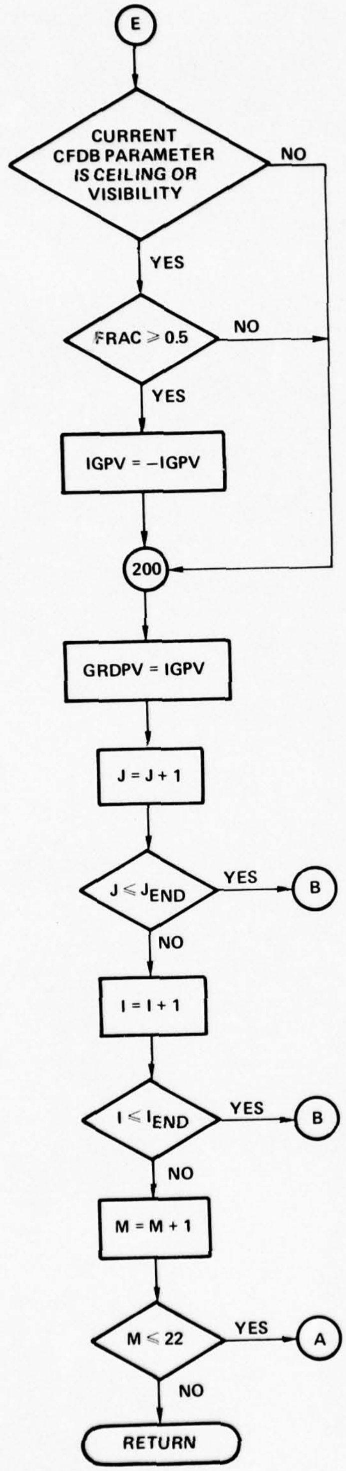
Calculate running sum of weight factor, SMWF, and weighted best report value of current CFDB parameter, SMWFO. OBS = current best report value for current CFDB parameter.



Jump back to C if there are more collected best reports for current CFDB parameter.

Calculate the fraction of collected best reports which were tagged, FRAC, and the weighted average of the best reports, GPV, for the current CFDB parameter. Note – These calculations are overridden in the case of present weather.

Code the integer weighted average of best report layered cloud covers as thin if a majority of these best reports were tagged as thin.



Code the integer weighted average of best report ceiling or visibility as variable if a majority of the best reports of the parameter were tagged as variable.

Set the current grid point value of the current CFDB parameter equal to IGPV.

Increment J grid point index.

Jump back to B if more grid points.

Increment I grid point index.

Jump back to B if more grid points.

Increment CFDB parameter index.

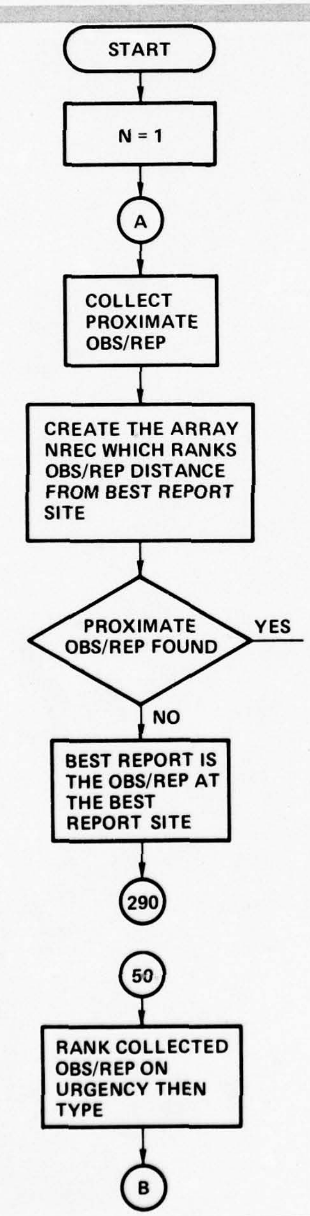
Jump back to A if more CFDB parameters.

SUBROUTINE COMOBR(NOB, DSP, TIME, LSFILE)

Ranks, resolves conflicting information, and combines CFDB elements of proximate OBS/REP'S: then insures internal consistency of combined OBS/REP.

Input Data

NOB = Number of time qualified OBS/REP
DSP = Maximum distance between OBS/REP to be combined into a best report, km.
TIME = Reference time or map time of CFDB creation or update.
LSFILE = Logical device No. of temporary storage file used.



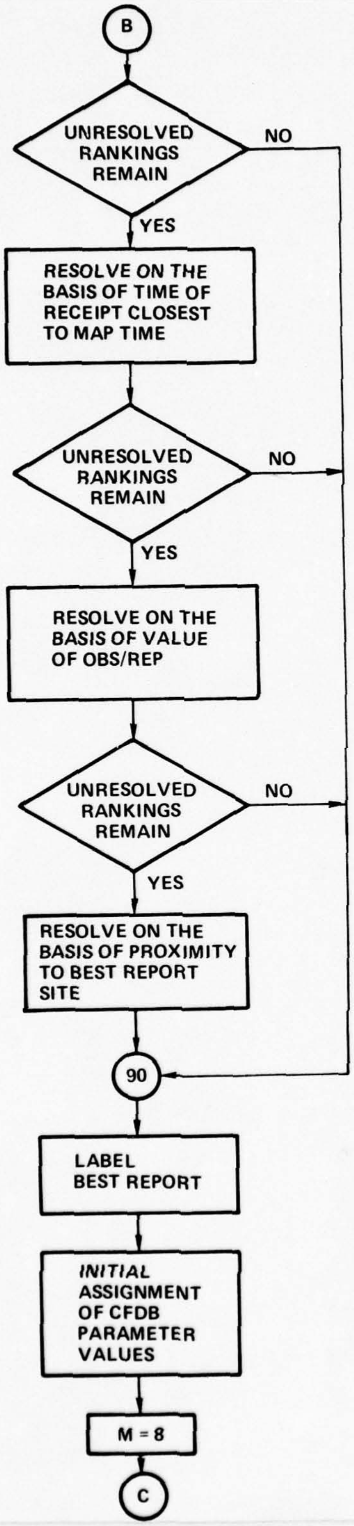
Initialize index of best report.

Scan the list of time qualified OBS/REP and collect ten or less which are within DSP km of the best report site.

Create a one dimensional array NREC, in which the collected proximate OBS/REP are ranked in order of increasing distance from the best report site.

Specials of all types out rank non specials. Types ranked as follows:

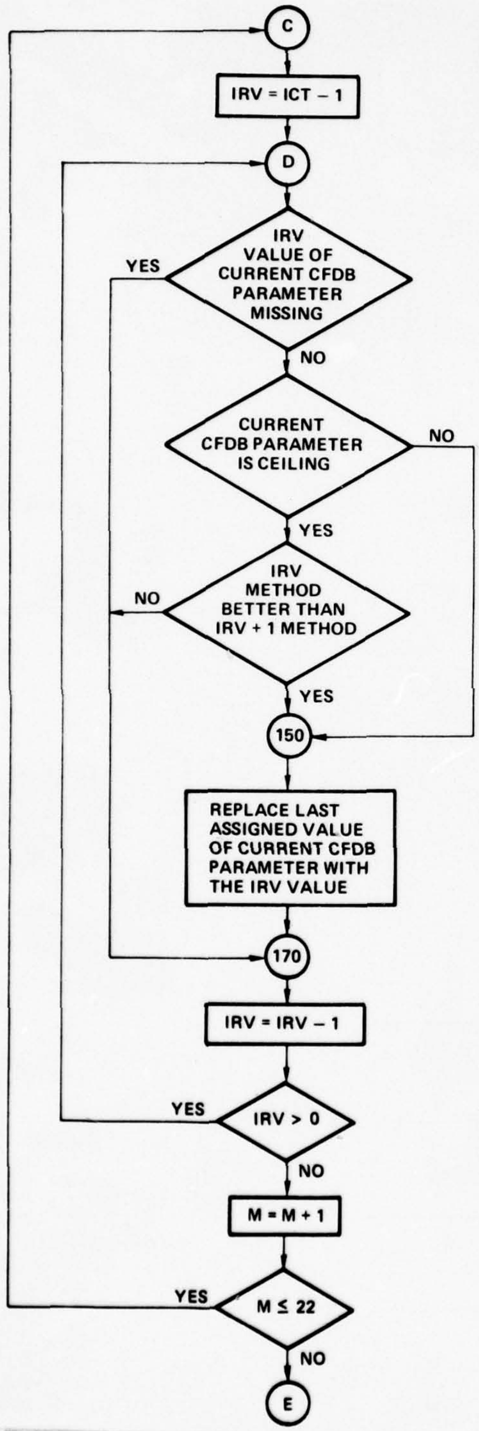
1 AIRWAYS
2 METAR
3 SYNOP
4 RAOB
5 AFGWC-3D NEPH PROG



Assign the location, station elevation, time sequence number and type of OBS/REP at the best report site to the best report.

Assign the CFDB parameters of the lowest ranking OBS/REP to the best report.

Initialize index of CFDB parameter.



Initialize counter to index number of second lowest ranking OBS/REP.

Jump to 170 if the IRV value of the current CFDB parameter is missing.

Jump to 150 if the current CFDB parameter is not ceiling.

Jump to 170 if the method of measuring the ceiling in the IRV OBS/REP is not better than the method used in the IRV + 1 OBS/REP. The hierarchy of ceiling measurements is:

First	– MEASURED
Second	– AIRCRAFT
Third	– BALLOON
Fourth	– RADAR
Fifth	– ESTIMATED
Sixth	– INDEFINITE

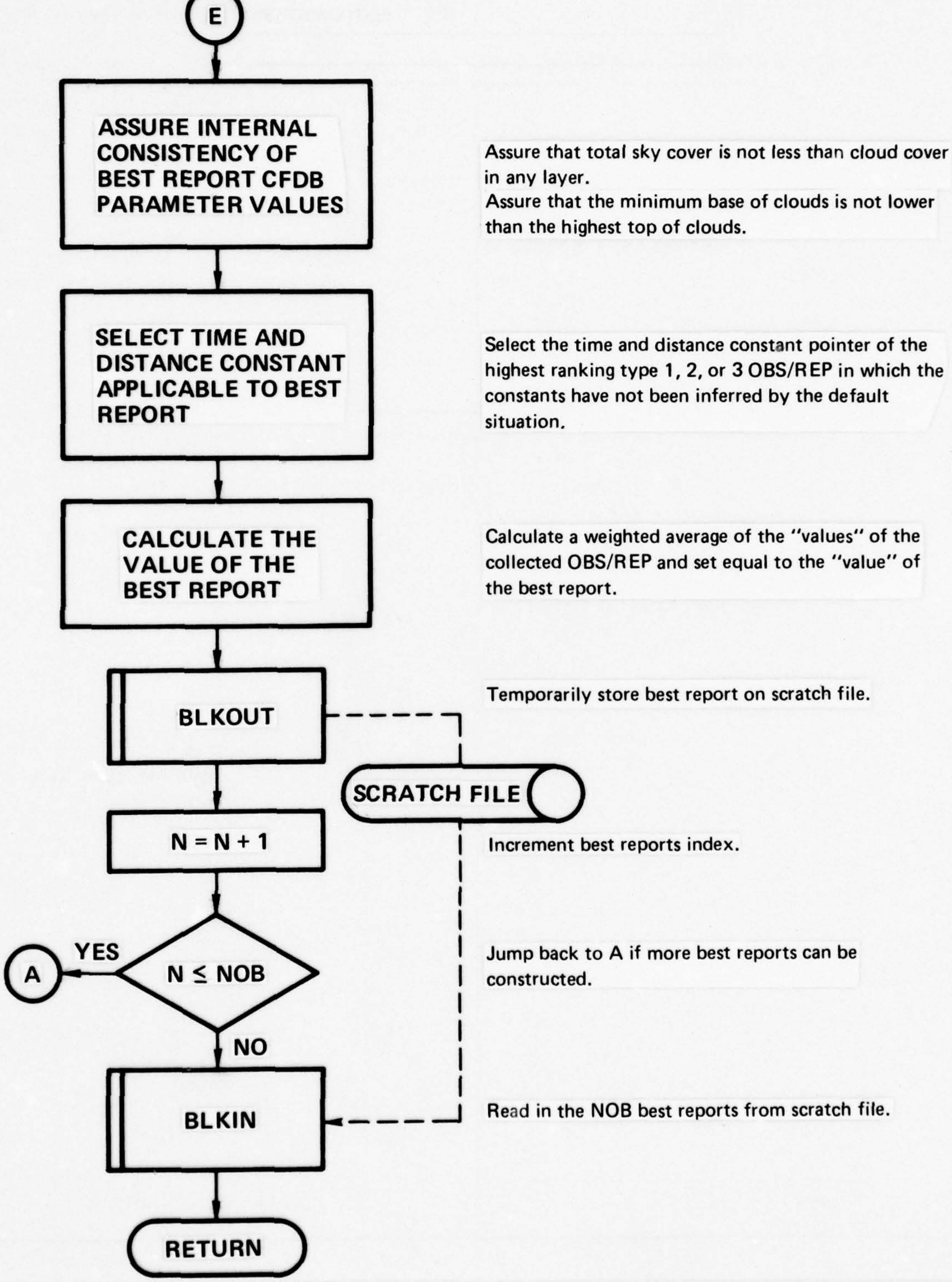
A replacement is not made if the IRV value of the current CFDB is missing and additionally in the case of ceiling when the method of measurement is not better than the method used for the value last assigned.

Decrement index of collected OBS/REP.

Jump back to D if there are more collected OBS/REP.

Increment index of CFDB parameter.

Jump back to C if there are more parameters.



E
ASSURE INTERNAL CONSISTENCY OF BEST REPORT CFDB PARAMETER VALUES
Assure that total sky cover is not less than cloud cover in any layer.
Assure that the minimum base of clouds is not lower than the highest top of clouds.
SELECT TIME AND DISTANCE CONSTANT APPLICABLE TO BEST REPORT
Select the time and distance constant pointer of the highest ranking type 1, 2, or 3 OBS/REP in which the constants have not been inferred by the default situation.
CALCULATE THE VALUE OF THE BEST REPORT
Calculate a weighted average of the "values" of the collected OBS/REP and set equal to the "value" of the best report.
BLKOUT
Temporarily store best report on scratch file.
SCRATCH FILE
N = N + 1
Increment best reports index.
YES
A
N ≤ NOB
Jump back to A if more best reports can be constructed.
NO
BLKIN
Read in the NOB best reports from scratch file.
RETURN

SUBROUTINE DEPCLD (PRES, TEMP, DEP, NCLD)

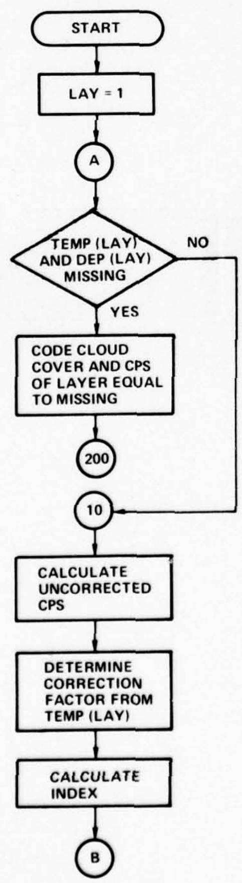
Routine to convert dewpoint depression, temperature, and pressure information into percent cloud cover.

CPCLD1 = CPS to cloud conversion table at 850 MB.
CPCLD2 = CPS to cloud conversion table at 700 MB.
CPCLD3 = CPS to cloud conversion table at 500 MB.
CPCLD4 = CPS to cloud conversion table at 300 MB.
PRESTD = Standard pressure levels for CPS to cloud conversion.
NCLD = Percent cloud cover
DPRCPS = Conversion factors for dewpoint depression
TCOR = Temperature correction for CPS
PRES = Midpoint pressure of CFDB layer, millibars
TEMP = Midpoint temperature of CFDB layer, deg. K
DEP = Midpoint dewpoint depression of CFDB layer, deg. C
A, B, C = Constants in the expression

DPRCPS = A+ B* (pressure/1000) + C* (pressure/1000) **2

This expression converts dewpoint depression to condensation pressure spread conversion factors for CFDB layers.

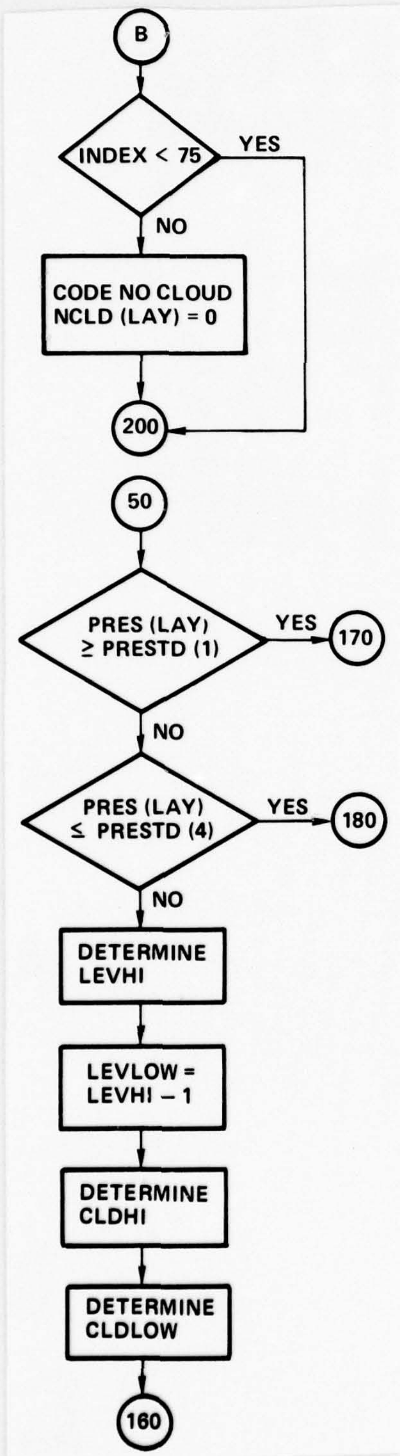
CPS = Condensation pressure spread of CFDB layers



Initialize CFDB layer index.

Jump to 10 if temperature and dewpoint depression of layer are not missing.

Determine appropriate entry in CPS to cloud table.



INDEX too large, no cloud possible.

Jump to 170 if the midpoint pressure of the CFDB layer is equal or greater than the pressure of the lowest table.

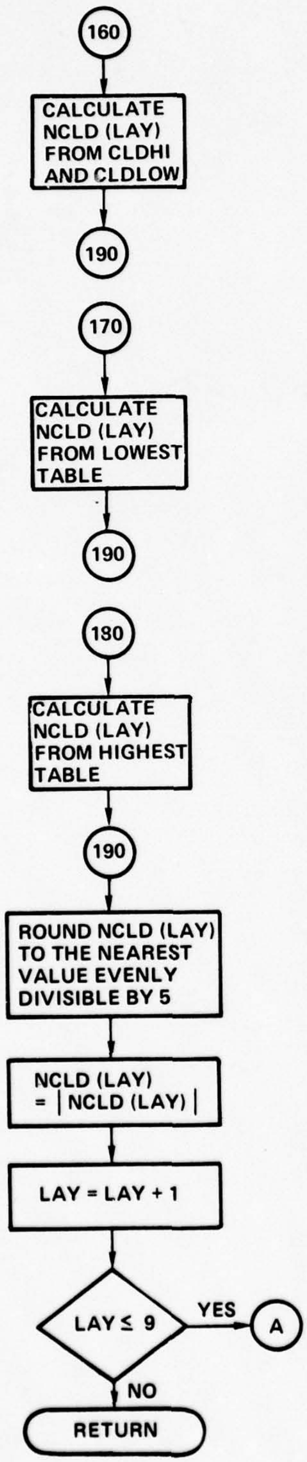
Jump to 180 if the midpoint pressure of the CFDB layer is equal to or less than the pressure of the highest table.

Determine LEVHI, the index number of the upper bound pressure level which is defined as the pressure level of the lowest table whose associated pressure is less than that of the midpoint of the CFDB layer.

Calculate the index number, LEVLOW, of the lower bound pressure level.

Determine a cloud cover, CLDHI, from the upper bound table.

Determine a cloud cover, CLDLOW, from the lower bound table.



Linearly interpolate with respect to pressure to calculate the cloud cover at the midpoint pressure of the CFDB layer from CLDHI and CLDLOW.

Determine cloud cover of the CFDB layer from the lowest table.

Determine cloud cover of the CFDB layer from the highest table.

Round cloud cover of the CFDB layer to the nearest 5 percent.

Guard against a minus zero value of cloud cover occurring in round off.

Increment CFDB layer index.

Jump back to A for more CFDB layers.

SUBROUTINE FIND1B (INCODE, IX, IY, RADIUS, ITMIN, ITMAX, *IREC, NOMORE)

FIND1B is used when the user wishes to examine all the OBS/REP's stored that are within a specified radius of specified coordinates which were observed during a specified time interval. Each call to FIND1B returns one OBS/REP going backward in time sequence.

INCODE = User control code. INCODE = 1 initiates the sequence and searches for the newest OBS/REP which satisfies the location and time requirements. This OBS/REP is returned to the user in user buffer IREC. INCODE NOT = 1 is used on *successive calls to retrieve the next* OBS/REP *in backward time sequence.*

IX = Relative X position in hectometers.

IY = Relative Y position in hectometers.

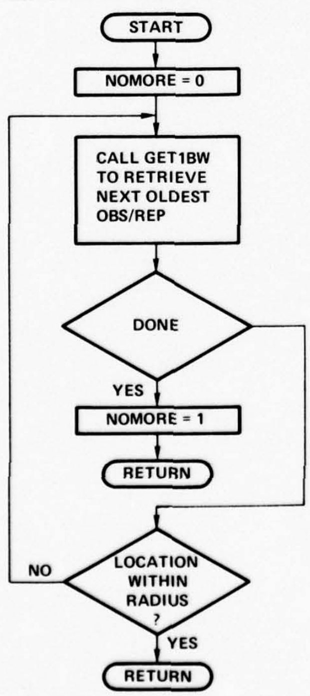
RADIUS = Radius in hectometers of circle to be centered at (IX, IY). All OBS/REP's returned to user will be in this circle.

ITMIN = Minimum, or oldest, observation time in minutes (0-1439).

ITMAX = Maximum, or newest, observation time in minutes (0-1439). FIND1B will return OBS/REP's starting at ITMAX, or older.

IREC = Buffer in calling routine containing NWDREC words where the OBS/REP will be stored.

NOMORE = Status returned to user. NOMORE = 0 indicates that an OBS/REP was returned to the user in IREC and that there may be more OBS/REP's if the user should call again. NOMORE = 1 indicates that no OBS/REP was returned and that no additional OBS/REP's exist in the data base within the specified time and location constraints. The user should assume that the contents of IREC will be modified whenever *FIND1B is called.*



Assume an OBS/REP will be found.

Going backward in observation time from ITMAX.

Processing is complete when either the oldest OBS/REP in file J is examined, or, the observation time of the OBS/REP returned by GET1BW is older than ITMIN.

If the location of the OBS/REP is within the specified radius of IX and IY, return the OBS/REP to the user buffer IREC.

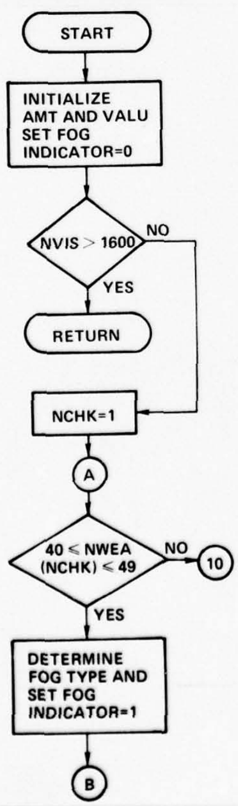
SUBROUTINE FOG (NVIS, NWEA, AMT, VALU)

Routine to check for fog and make decisions as to percentage cloud cover and tops of clouds based on horizontal visibility and type of fog.

NVIS = horizontal visibility in meters
NWEA = surface weather WMO code 4677
Derived layered cloud information
NUMLAY = number of layers generated
KIND = kind of cloud layer
- 1 = low
- 2 = middle
- 3 = high
- 4 = fog
- 5 = lowest cloud
- 6 = clear layer

ITHIN = thin layer designator
- MISSING = not thin
- 1 = thin

COVER = cloud cover in layer (0.0 – 1.0)
BASE = height of the base of layer, feet
TOP = height of top of cloud layer, feet
AMT = cloud cover due to fog
VALU = value of OBS/REP



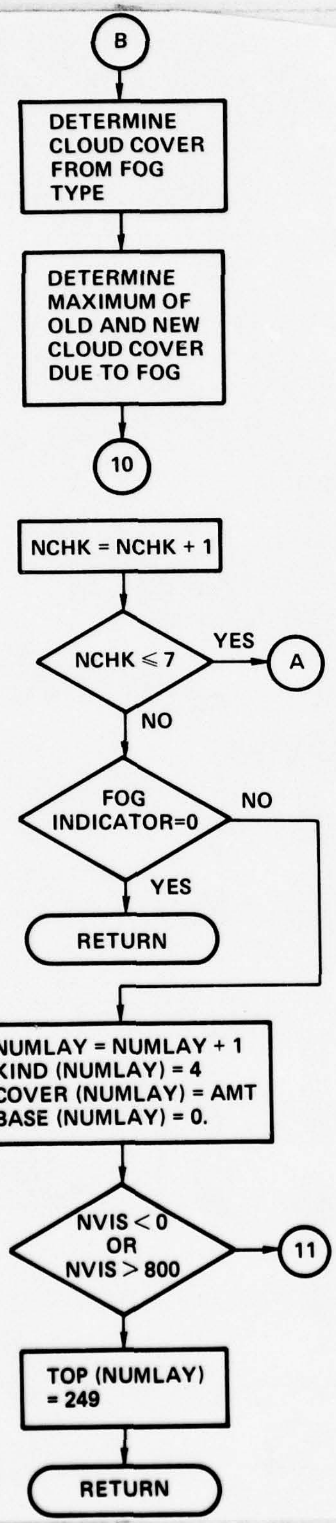
Set AMT = 0. Set VALU = 10 if there was no layered cloud data present in OBS/REP. If there was layered cloud data present then set VALU = (VALU + 10.)/2.

Return if visibility is greater than 1600 meters (1 mile).

Initialize counter for surface weather array, NWEA.

Jump to 10 if surface weather does not show the presence of fog.

Determine the fog type from the surface weather code.

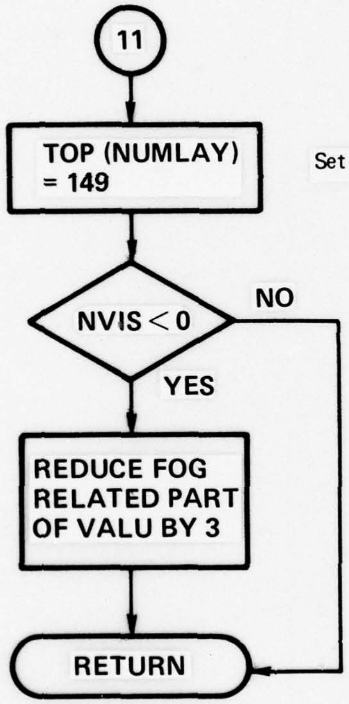


Increment counter for surface weather array.

Increment layer counter, set cloud cover and base.

Jump to 11 if horizontal visibility is unknown or greater than 800 meters (1/2 mile)

Set top of fog layer equal to 249 feet.



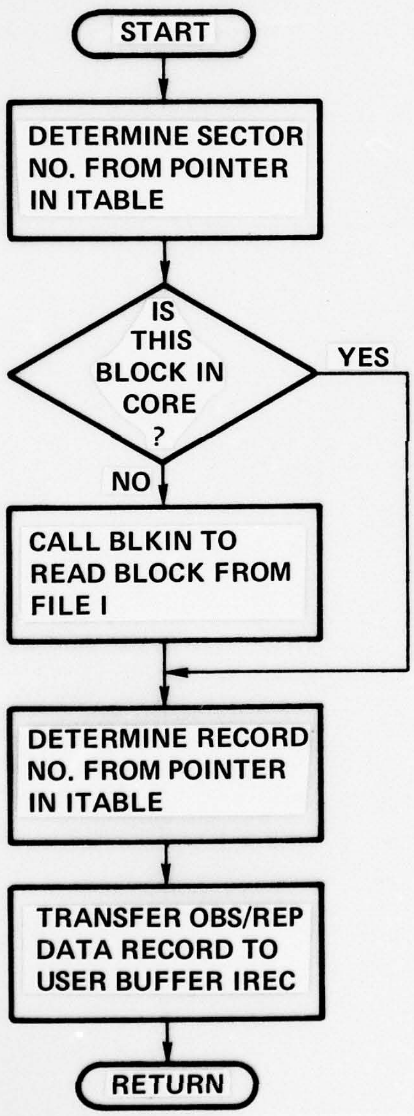
Set top of fog layer equal to 149 feet.

SUBROUTINE GETOBI (ITABID, IREC)

Get an OBS/REP from file I.

ITABID = Column index of ITABLE pointing to desired OBS/REP.

IREC = Buffer in user program where OBS/REP will be stored.



Sector No. corresponds to block No. in file I.

IBLOCK in COMMON/BASE/ equals the No. of the file I block in core buffer IBUF.

Mass storage to core transfer.

Record No. in block.

SUBROUTINE GET1BW (INCODE, NTIME, IREC, NOMORE)

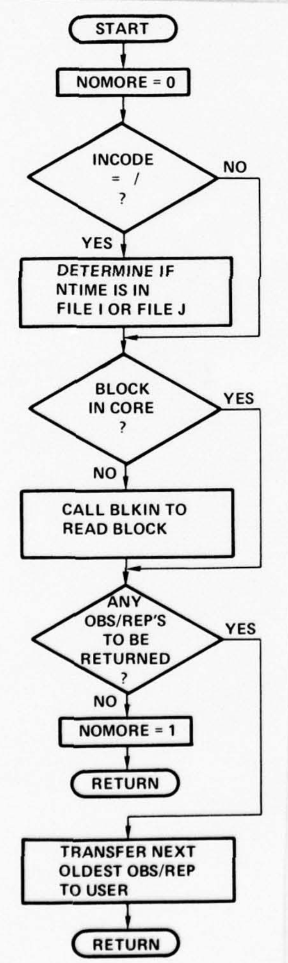
GET1BW is used when the user wishes to examine all the OBS/REP's stored starting at NTIME and going backward in time sequence.

INCODE = User control code. INCODE = 1 initiates the sequence and searches for the first record which is returned to the user. INCODE NOT = 1 is used in successive calls to retrieve the next OBS/REP in time sequence.

NTIME = Start time in minutes (0-1439).

IREC = Buffer in calling routine containing NWDREC words where the OBS/REP will be stored.

NOMORE = Status returned to user. NOMORE = 0 indicates that an OBS/REP was returned to the user in IREC and that there may be more OBS/REP's if the user should call again. NOMORE = 1 indicates that no OBS/REP was returned and that no additional OBS/REP's exist in the data base.



Assume an OBS/REP will be returned.

If the time sequence starts in file I, and there are additional calls, older data records will be extracted from file I and then from file J.

Read proper block from file I or file J only if not in core at this time.

The OBS/REP returned on the previous call was the oldest OBS/REP in the data base.

Transfer to buffer starting at IREC.

SUBROUTINE GET1FW (INCODE, NTIME, IREC, NOMORE)

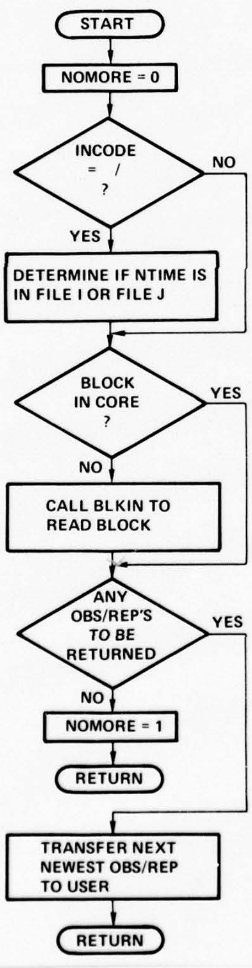
GET1FW is used when the user wishes to examine all the OBS/REP's stored starting at NTIME and going forward in time sequence.

INCODE = User control code. INCODE = 1 initiates the sequence and searches for the first record which is returned to the user. INCODE NOT = 1 is used on successive calls to retrieve the next OBS/REP in time sequence.

NTIME = Start time in minutes (0-1439).

IREC = Buffer in calling routine containing NWDREC words where the OBS/REP will be stored.

NOMORE = Status returned to user. NOMORE = 0 indicates that an OBS/REP was returned to the user in IREC and that there may be more OBS/REP's if the user should call again. NOMORE = 1 indicates that no OBS/REP was returned and that no additional OBS/REP's exist in the data base.



Assume an OBS/REP will be returned.

If the time sequence starts in file J, and there are additional calls, newer data records will be extracted from file J and then from file I.

Read proper block from file J or file I only if not in core at this time.

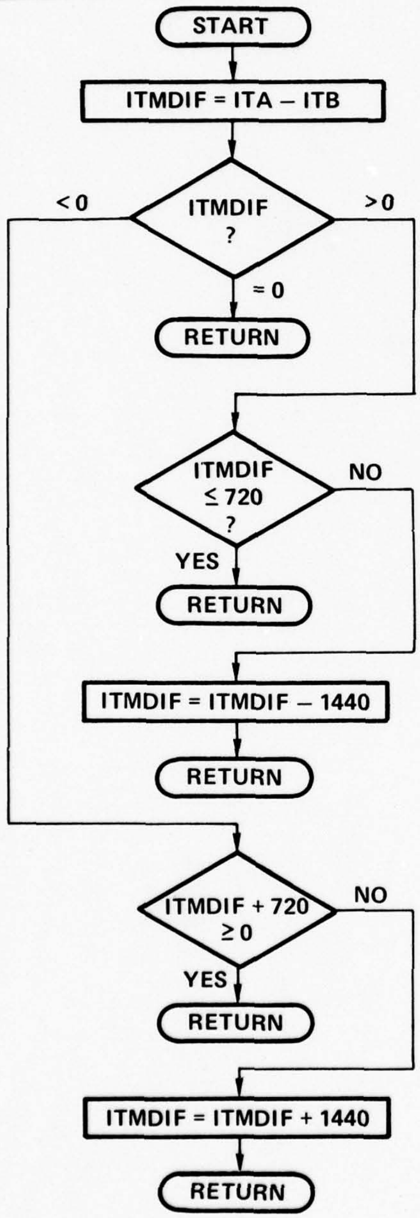
The OBS/REP returned on the previous call was the most recent OBS/REP in the data base.

Transfer to buffer starting at IREC.

FUNCTION ITMDIF (ITA, ITB)

Computes difference between times ITA and ITB. Result is positive if ITA is more recent than ITB. It is assumed that all time differences will be less than or equal to 720 minutes.

ITA and ITB are time values in minutes (0-1439).



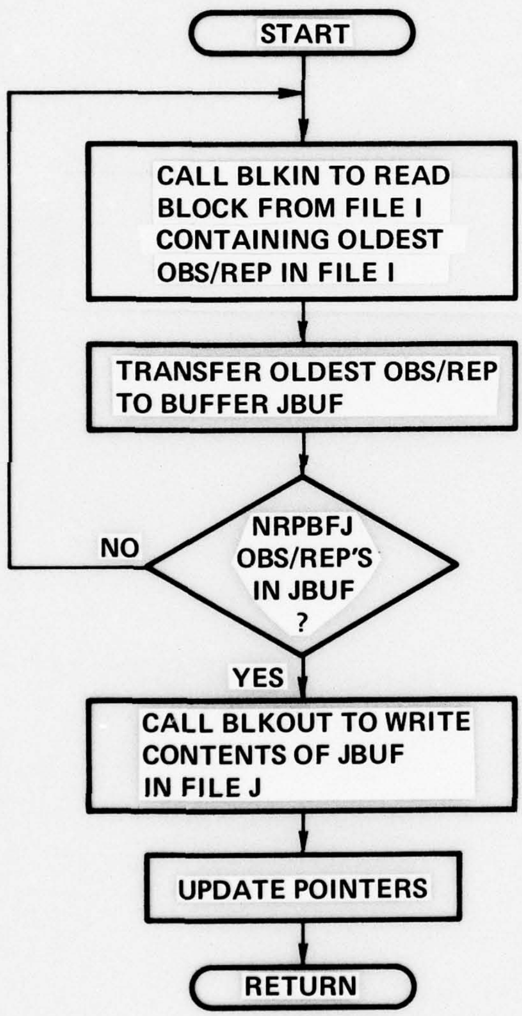
Take difference.

ITA must be older than ITB.

ITA must be more recent than ITB.

SUBROUTINE ITOJ

Delete the oldest (NRPBFJ) records from file I and store them as a block in file J.



Mass storage to core transfer.

Blocks in file J contain OBS/REP's sorted in terms of observation time.

NRPBFJ = the number of data records per block in file J and is defined in subroutine BEGIN.

Core to mass storage transfer.

The NRPBFJ oldest 4 word entries in ITABLE no longer used.

SUBROUTINE LAYCLD (DLAT, VALU)

Routine to construct cloud layers from layered cloud data in AIRWAYS, METAR, and SYNOP type OBS/REP.

List of Arguments

Input

DLAT = Latitude of OBS/REP, degrees (negative if south)

Output

VALU = Information VALU of OBS/REP

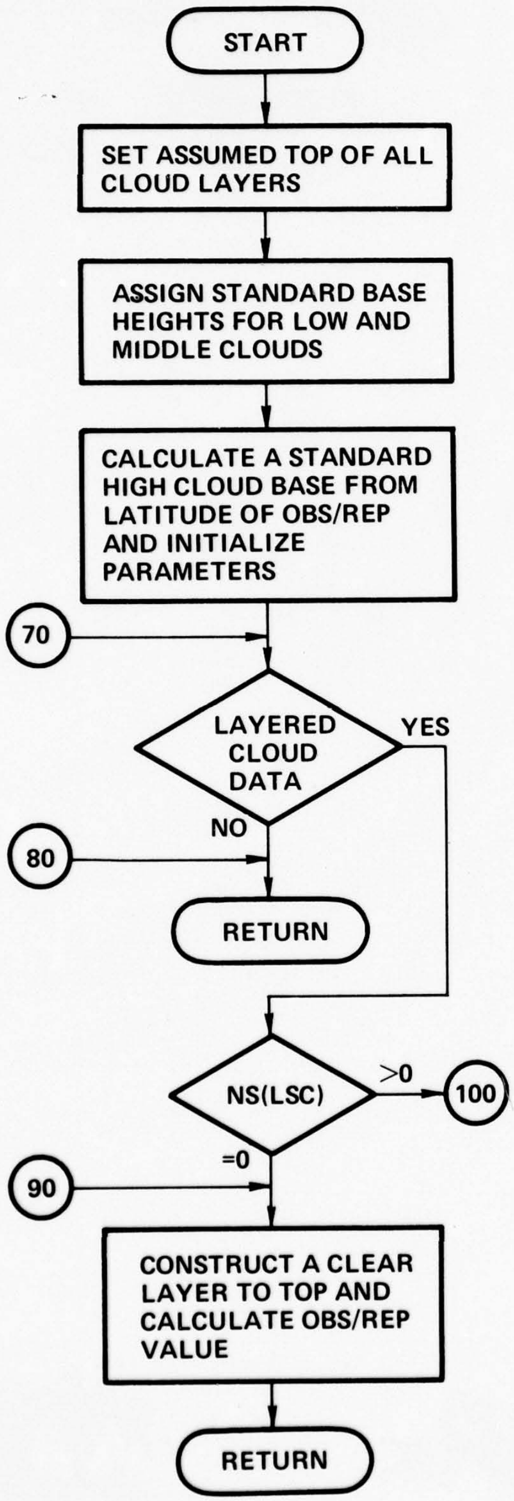
Common Data

In

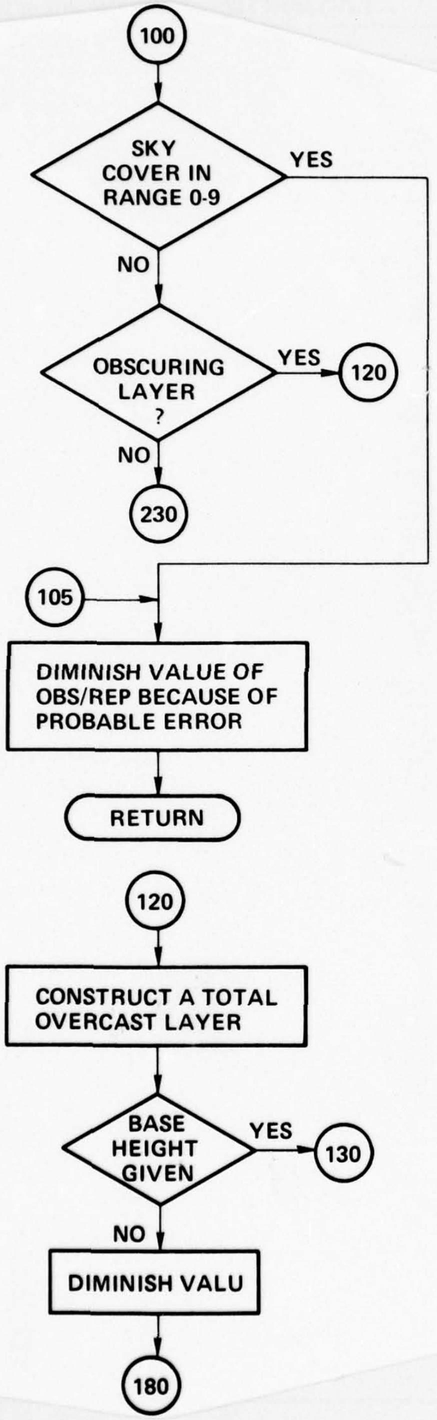
NS(J) = Sky cover due to cloud in layer, 0-9. 1 to 10 layers.
ICTS = Type of cloud in layer, 0-9 WMO code 0500
IHS(J) = Height of base of cloud layer
  AIRWAYS – 100's of feet
  METAR – WMO code 1677
  SYNOP – WMO code 1677
ITHIN(J) = Cloud layer thickness indicator
  1 if thin
  Missing if not thin
ITYPE = Type of OBS/REP
  1 = AIRWAYS   -1 if a special
  2 = METAR   -2 if a SPECI (special)
  3 = SYNOP

OUT

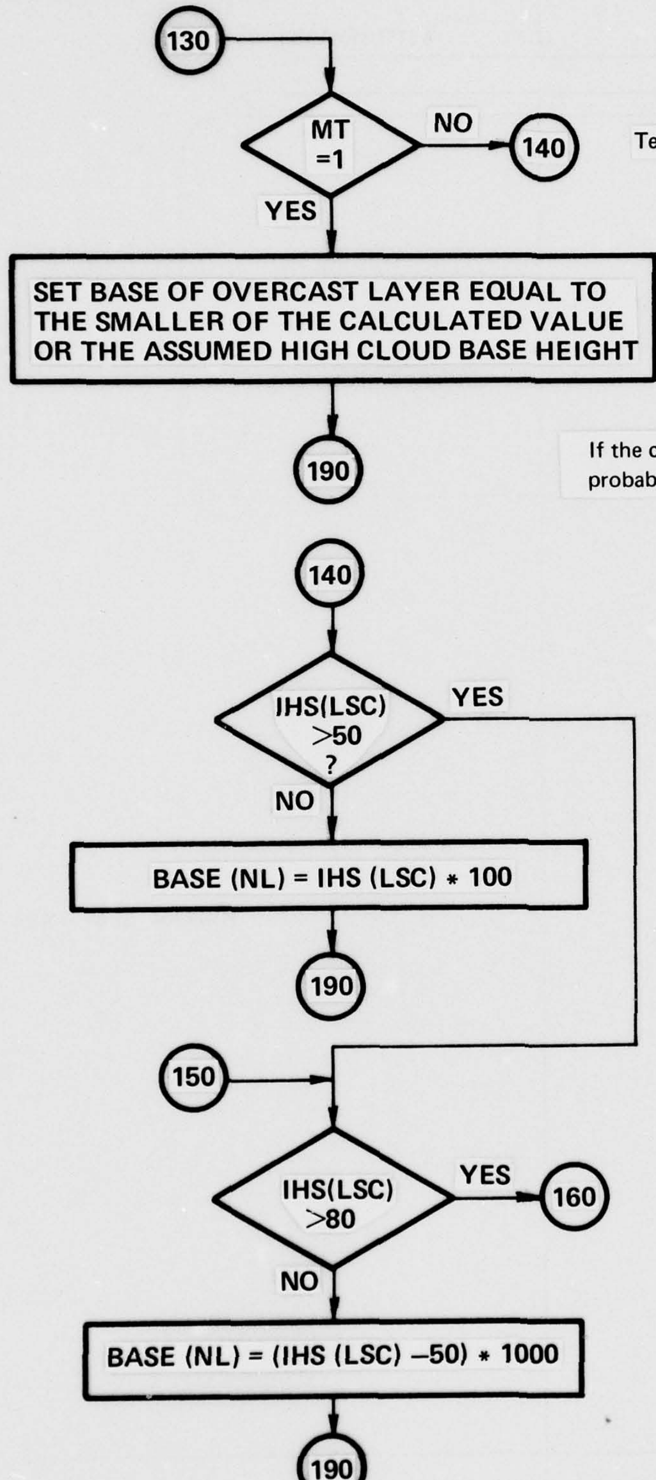
NUMLAY = Number of cloud layers identified
KIND = Kind of cloud layer
  1 = Low
  2 = Middle
  3 = High
  4 = Fog
  5 = Lowest cloud
  6 = Clear layer
ITHIN = Thin layer designator
MISSING = Not thin
  1 = Thin
COVER = Fraction of sky covered by clouds in the layer (0.0 - 1.0)
BASE = Height of the base of cloud layer, feet.
TOP = Height of the top of the cloud layer, feet.



Return if NS (LSC) is less than 0

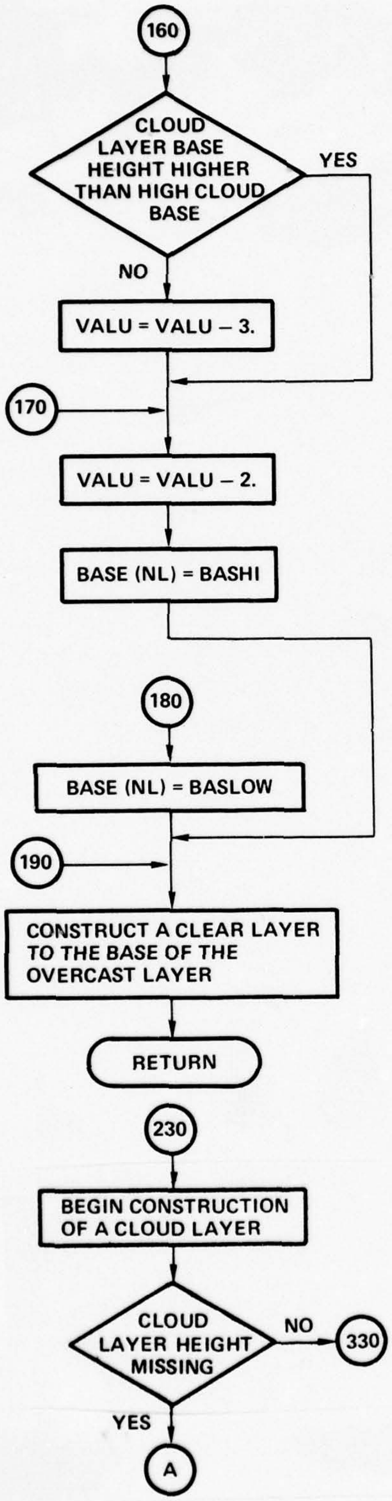


VALU of OBS/REP is diminished because an out of range cloud cover is assumed to be a result of a communications or coding error of a valid observation of the presence or absence of a cloud layer.



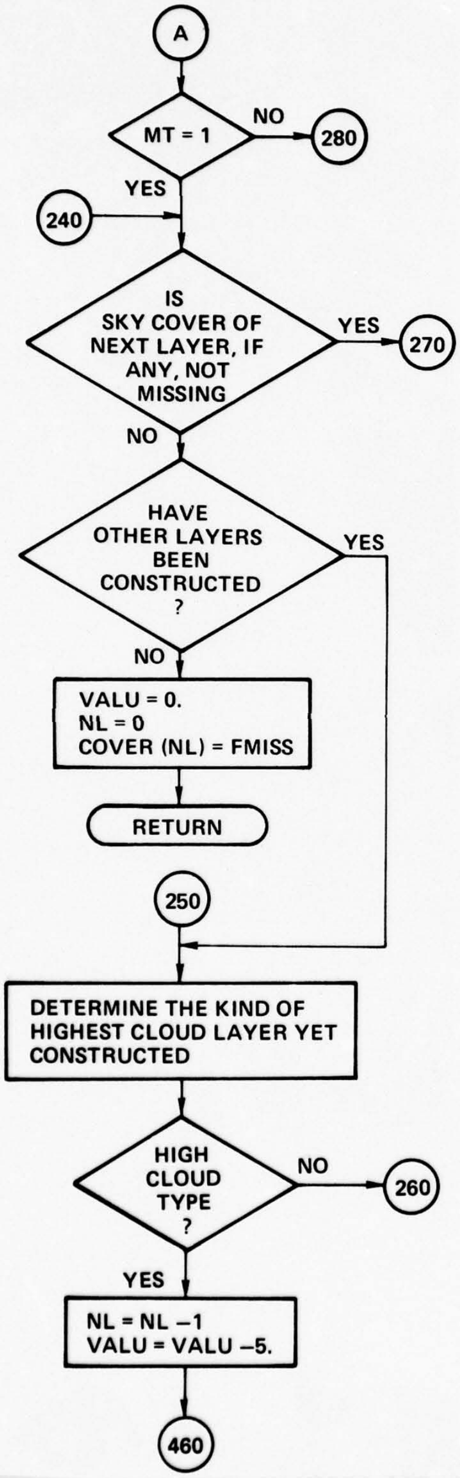
Test for AIRWAYS type OBS/REP

If the calculated value was larger IHS(LSC) was probably incorrect.



Cloud layer base height out of allowable range – probable error. Reduce VALU by a total of 5 and use the standard high cloud base.

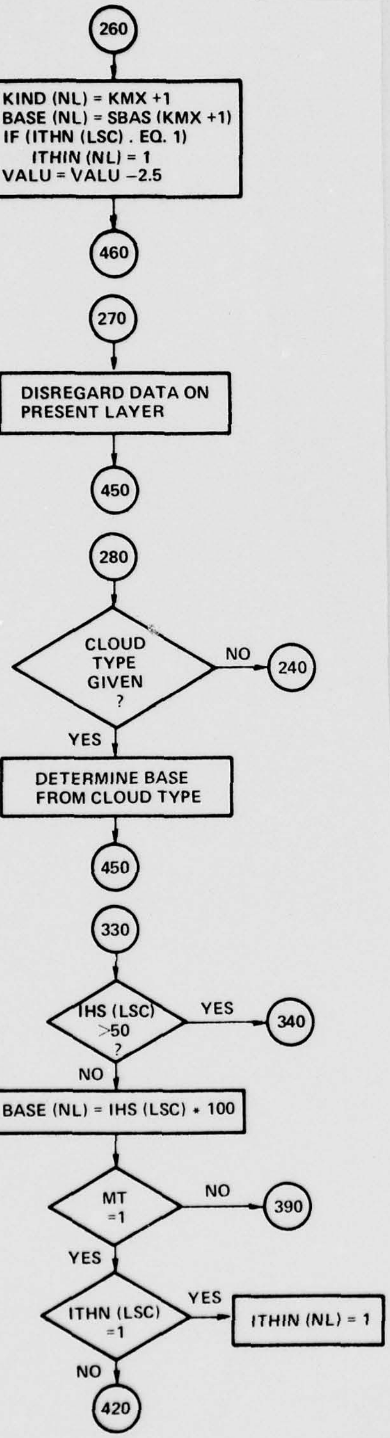
Comes here if not an obscuring layer.



Jump to 280 for SYNOP or METAR coded OBS/REP and determine base height of cloud layer from cloud type.

1, 2, or 3

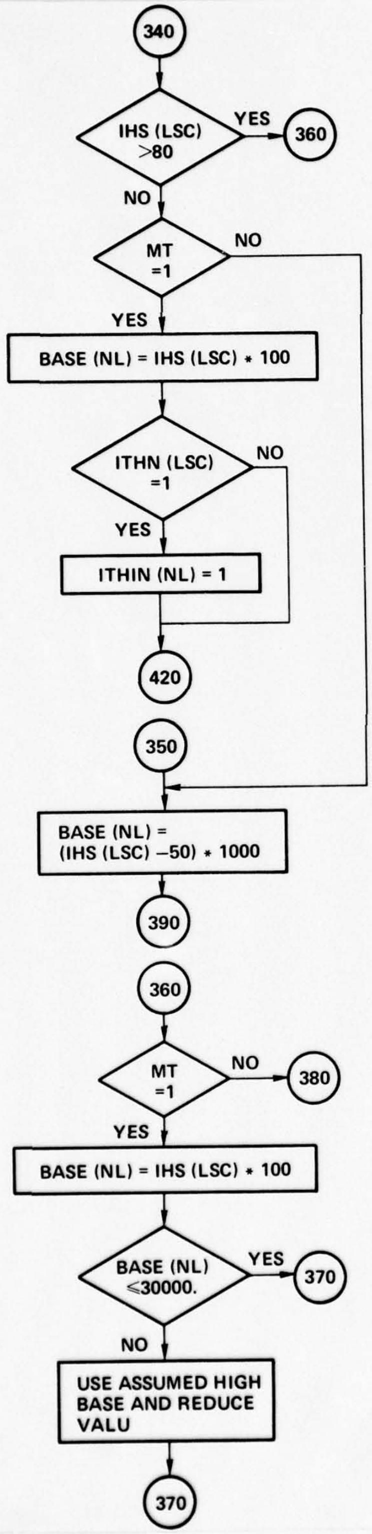
Probable error in data. Disregard present layer and reduce VALU.



Kind of highest cloud layer is 1 or 2.

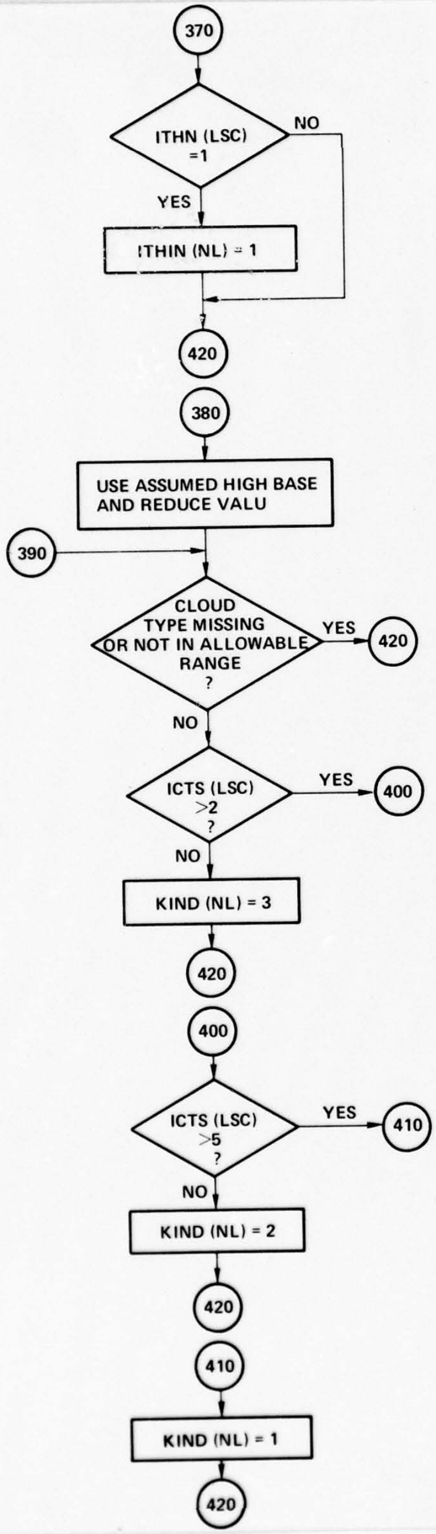
METAR and SYNOP OBS/REP with missing base heights come here.

Come here if base height code is not missing.



Go to 420 to determine kind of layer from base height.

Probable error in OBS/REP.

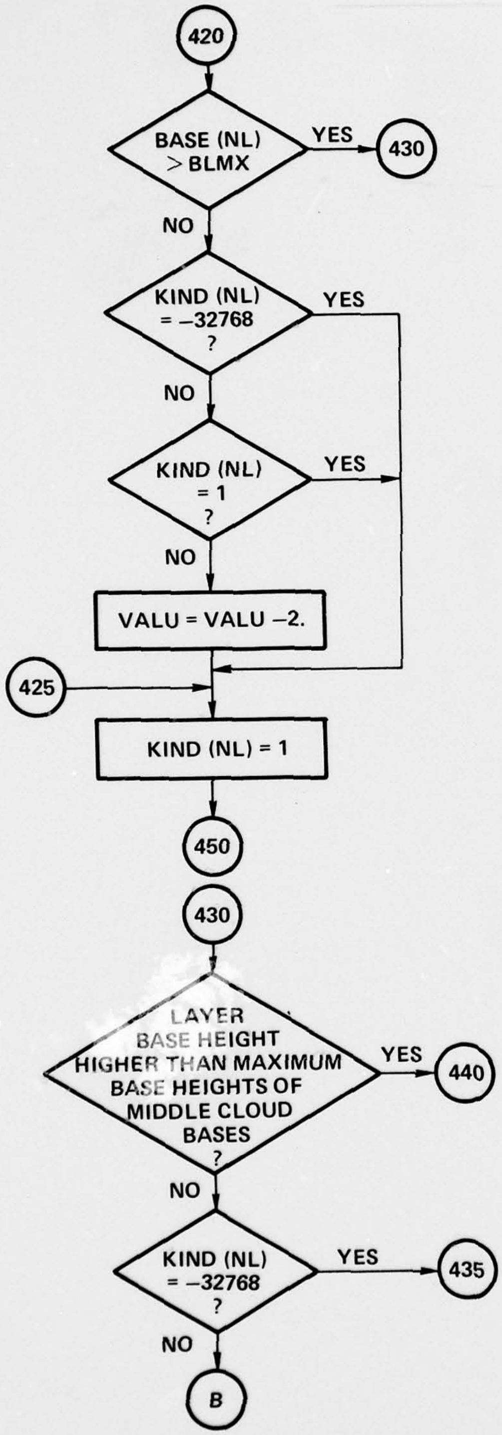


Probable error in OBS/REP.

Code layer high.

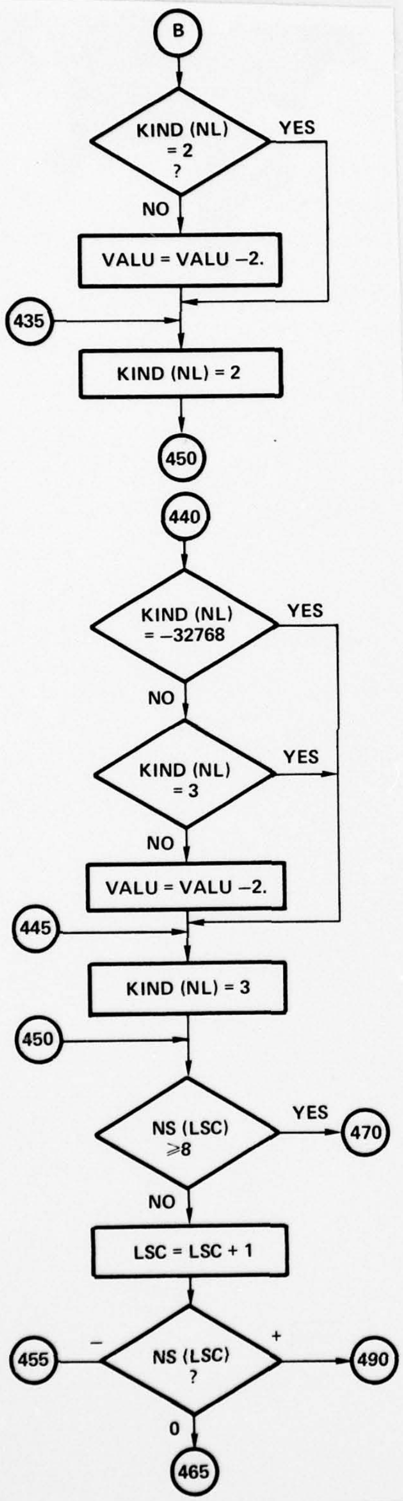
Code layer middle.

Code layer low.



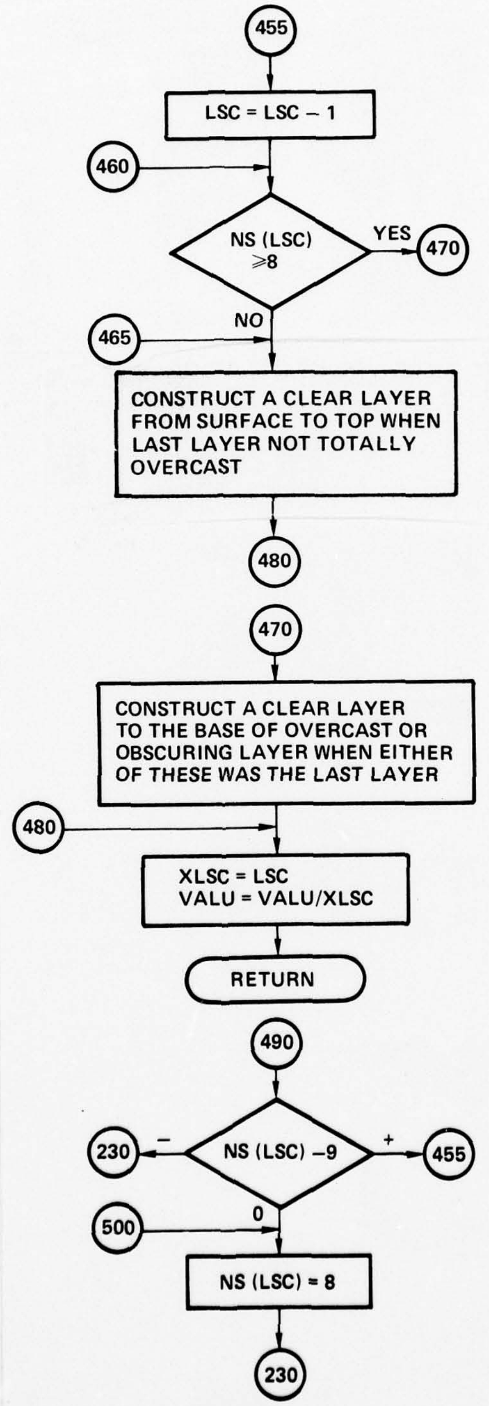
AIRWAYS and SYNOP or METAR OBS/REP with *missing cloud types come here to determine layer* kind. Also come here to check layer kind as determined from cloud type. Layer kind as determined from base height overrides determination from cloud type. Reduce VALU by 2. If the two determinations of kind do not agree.

Code layers low.



Code layer middle.

Test for overcast present layer, if not, test for more layered cloud data.



455
LSC = LSC – 1
460
NS (LSC) ⩾8
YES
470
NO
465
CONSTRUCT A CLEAR LAYER FROM SURFACE TO TOP WHEN LAST LAYER NOT TOTALLY OVERCAST
480
470
CONSTRUCT A CLEAR LAYER TO THE BASE OF OVERCAST OR OBSCURING LAYER WHEN EITHER OF THESE WAS THE LAST LAYER
480
XLSC = LSC
VALU = VALU/XLSC
RETURN
490
230
–
NS (LSC) –9
+
455
0
500
NS (LSC) = 8
230

SUBROUTINE MVLCOV (LCOVA, LCOVB, IHA, IHB)

This routine calculates the cloud cover in the CFDB layers of a station 'A', LCOVA(I), at an elevation of IHA (meters) that would exist if the layered cloud coverage at a station 'B', LCOVB(I), of elevation IHB (meters) were moved to 'A' with the CFDB layers of 'B' retaining their reference level, IHB.

Input data

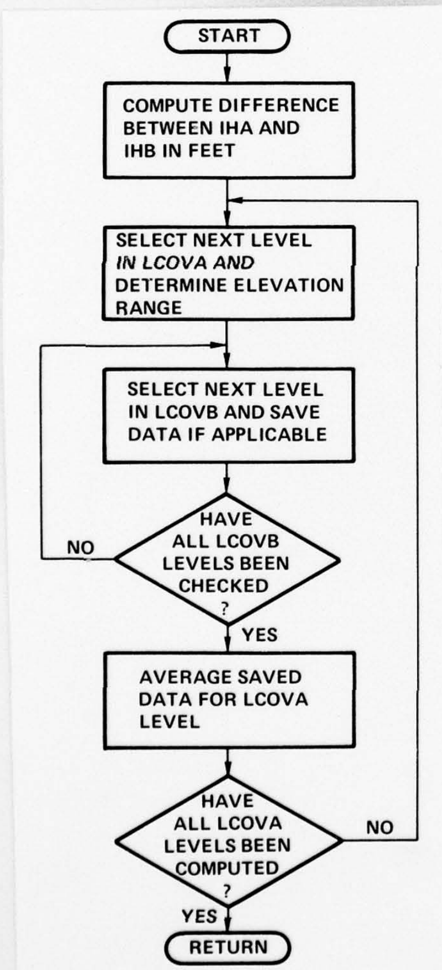
LCOVB(I) = Cloud cover in the CFDB layers of station 'B'

IHB = Height above mean sea level of station 'B'.

IHA = Height above mean sea level of station 'A'.

Output data

LCOVA(I) = Cloud cover in the CFDB layers of station 'A'.



CFDB layers are in multiples of feet.

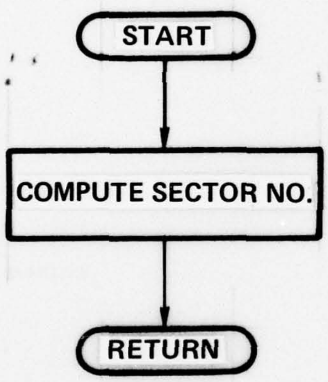
Process 9 layers.

–32768 indicates that no data exists.

FUNCTION NOSECT (IX, IY)

Computes sector No. (1-NSECTR) from UTM coordinates (IY, IX).

IX and IY are relative UTM coordinates.



The sector map is defined by subroutine SECTOR *using variables defined in subroutine* BEGIN.

SUBROUTINE RAOB (HMP, PMP, TMP, DMP, VALU)

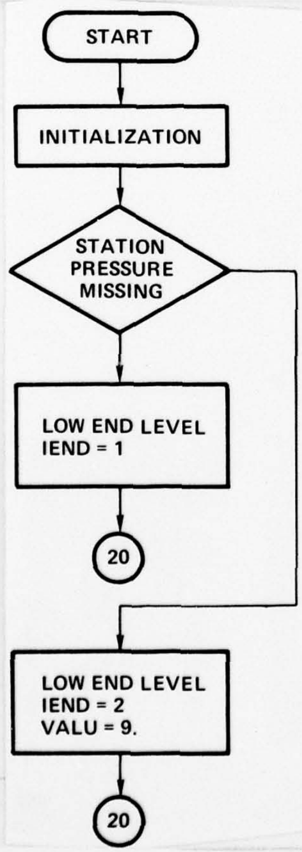
Routine to calculate temperature, dewpoint depression, and pressure for the midpoint of the CFDB layers.

Input Data

IX	=	X distance of RAOB site from IXREF, hectometers.
IY	=	Y distance of RAOB site from IYREF, hectometers.
IH	=	terrain height at RAOB site, meters.
ITIME	=	time of RAOB (0-1439).
ITYPE	=	4, (-4 if a special RAOB)
IZ(I)	=	altitude of RAOB reporting level, dekameters.
IP(I)	=	pressure of RAOB reporting levels, millibars*10.
IT(I)	=	temperature of RAOB reporting level, (deg. K.)*10.
IDD(I)	=	dewpoint depression of RAOB reporting level, (deg. C)*10.
NRRL	=	number of RAOB reporting levels
HMP(J)	=	height above mean sea level of midpoint of CFDB layers, meters.
PMP(J)	=	pressure at midpoint of the CFDB layers, millibars.
TMP(J)	=	temperature at midpoint of the CFDB layers, deg. K.
DMP(J)	=	dewpoint depression at midpoint of the CFDB layer, deg. K.

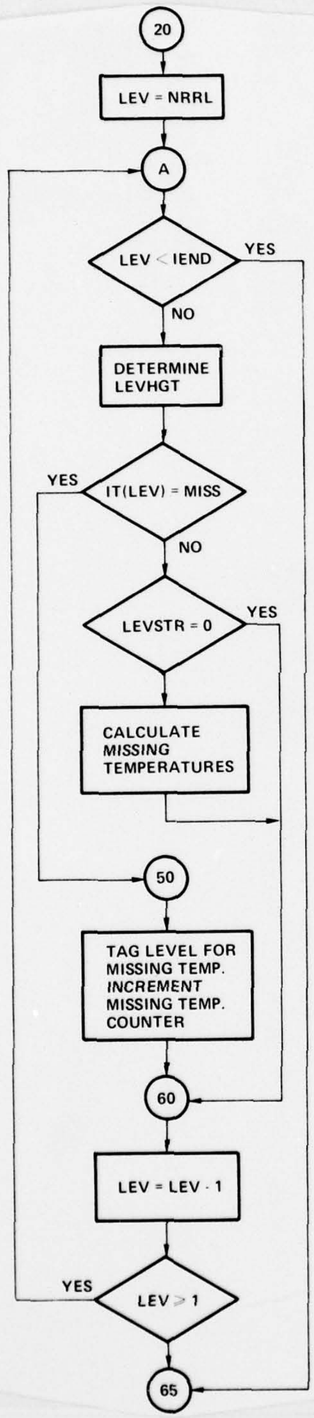
This Routine Assumes*

1. Pressures are in decreasing order
2. Station elevation is given
3. Temperature at top RAOB level is given
4. Temperature at two RAOB levels are given
5. First RAOB level is at surface
6. All pressures (except surface) are given
7. Missing data words are filled with -32768



Convert input integer altitude, pressure and temperature to floating point. Set initial value of VALU = 10.

Reduce VALU to 9. because of missing station pressure.



Initialize level index to highest level no.

Jump to 65 if current level no. is below low end level no.

Determine LEVHGT, the no. of the lowest level for which a height was reported.

Jump to 50 if temperature at current level is missing (MISS = - 32768)

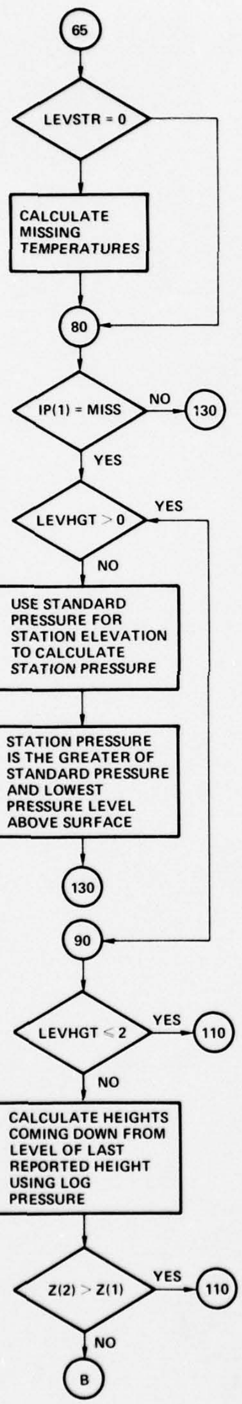
Jump to 60 if there currently are no levels at which temperatures were originally missing and which have not yet been calculated. If there are levels for which temperatures have to be calculated LEVSTR $\neq 0$.

Use log pressure interpolation to calculate the temperatures at the levels between the current level and the last level at which temperature was not missing.

Tag current level as the low end missing temperature level. If current level is also the first missing temperature level to be encountered following a non missing temperature, tag it as the high end missing temperature level.

Decrement level index by 1

Jump back to A if more levels remain.



Jump to 80 if there are no levels for which temperatures must be computed.

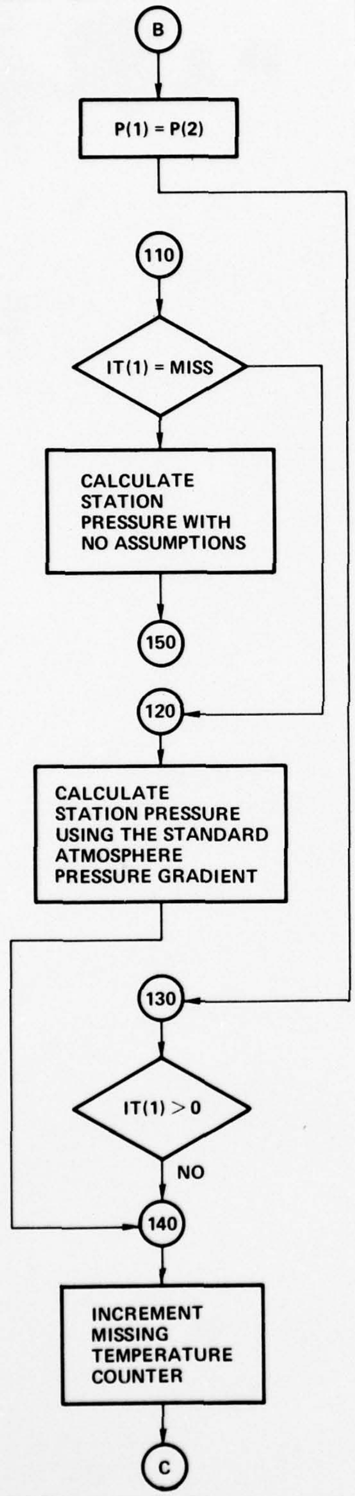
Use log pressure extrapolation to calculate missing temperatures at levels near the surface.

Jump to 130 if station pressure is not missing.

Jump to 90 if any heights of RAOB reporting levels were given.

Jump to 110 if a height was given for the lowest pressure level above the surface.

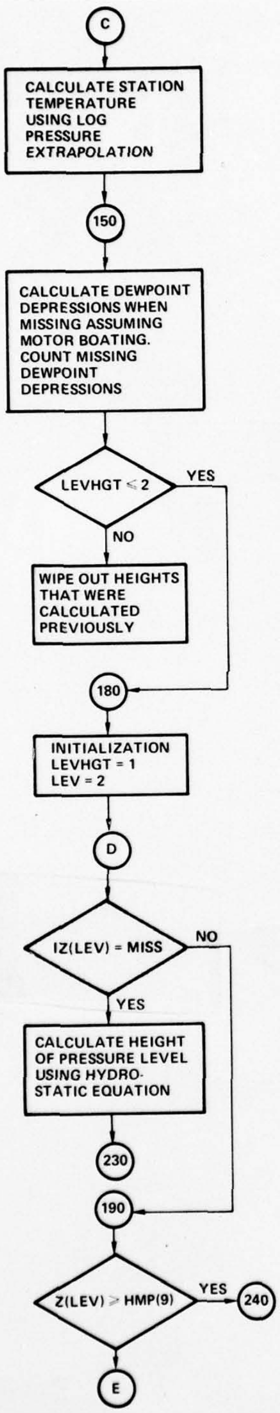
Jump to 110 if calculated height of second RAOB level is above calculated surface height.



Station pressure is the same as pressure of lowest RAOB level.

Jump to 120 if station temperature is missing.

Jump to 150 if station temperature is not missing.

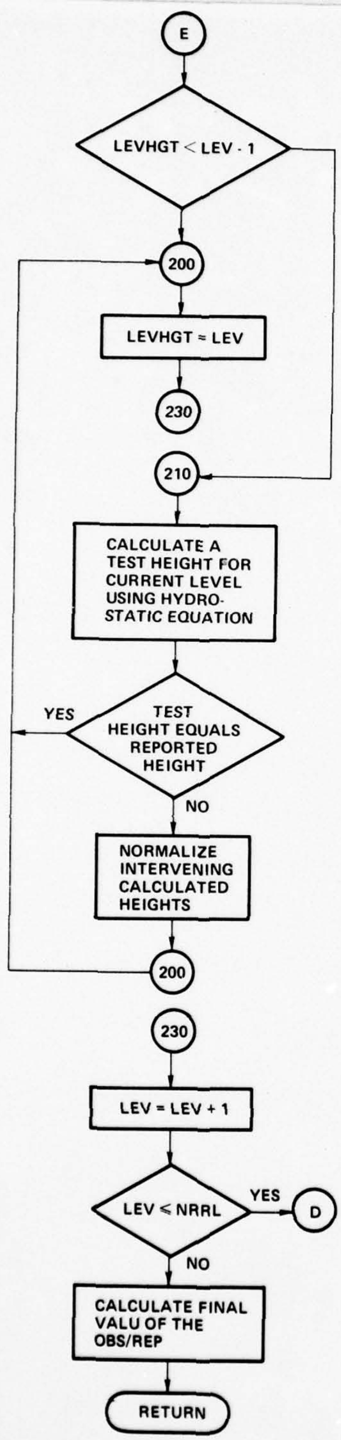


Jump to 180 if lowest level with a reported height was the first level above the surface level.

LEVHGT is a tag used to denote the current highest level at which there was a reported value of height.

Jump to 190 if reported height was missing.

Jump to 240 if height of pressure level is above the height of the midpoint of the highest CFDB layer.



If reported heights were missing at one or more levels before the current level, LEV, jump to 210.

Tag the current level as having a reported height.

Normalize the intervening calculated heights between the current level and the last level with a reported height on the actual height interval between these levels.

Jump back to D if there are more levels.

$$V_f = V_i - 4\left(\frac{M_T - M_{DD}}{N_L}\right)$$ where

V_f = final value

V_i = initial value

M_T = number of levels with missing temperatures

M_D = number of levels with missing dewpoint depressions

N_L = total number of levels.

SUBROUTINE RETOBR (INCODE, NTIME, INOBEL, NOMORE, TYMOLD)

This routine retrieves an OBS/REP from the file and checks for the presence or probability of convective type clouds.

Input Data

INCODE = user control code. INCODE = 1 initiates the sequence and searches for the first record which is returned to the user. INCODE NOT = 1 is used on successive calls to retrieve the next OBS/REP in time sequence.

NTIME = start time in minutes (0-1439).

TYMOLD = time of oldest OBS/REP to be retrieved

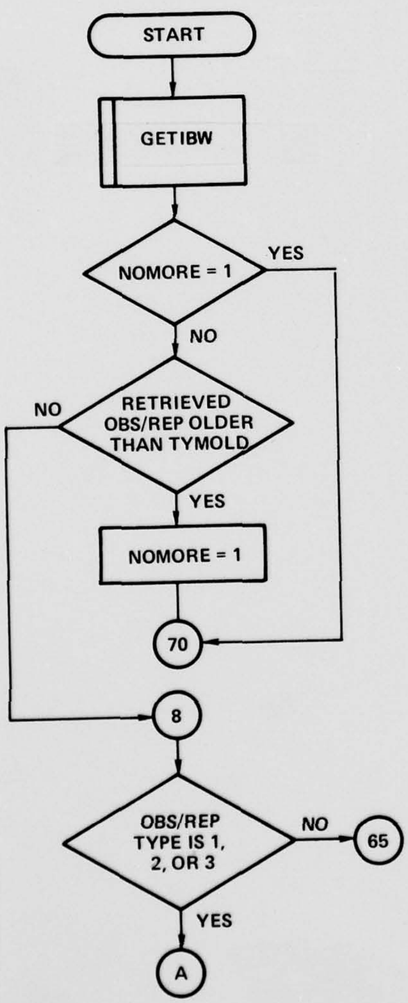
Output Data

INOBEL = retrieved OBS/REP

NOMORE = control code

0 = more OBS/REP on file

1 = no more OBS/REP on file or remainder of OBS/REP on file are older than TYMOLD

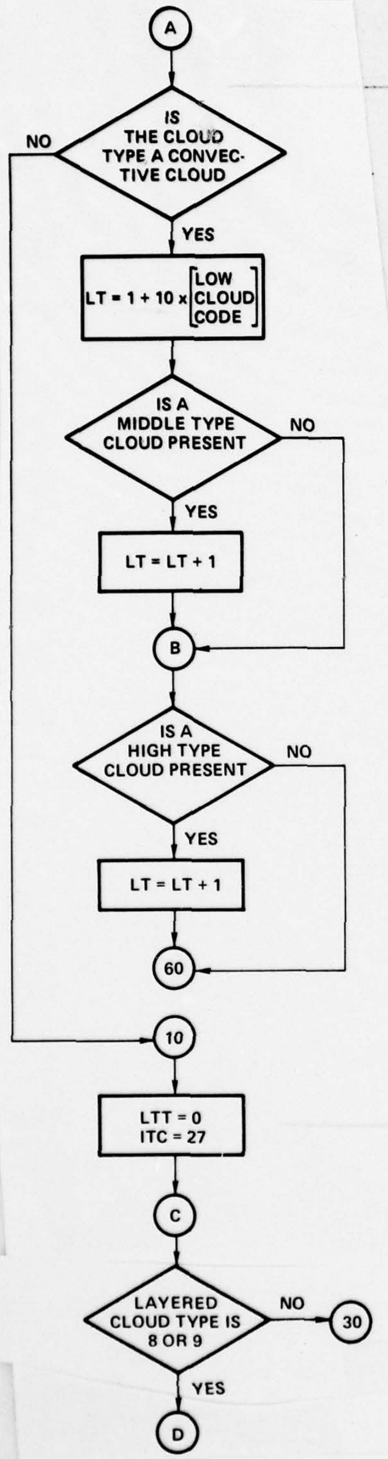


Retrieve an OBS/REP from the file.

Jump to 70 if there are no more OBS/REP on the file.

Jump to 8 if OBS/REP is not older than TYMOLD

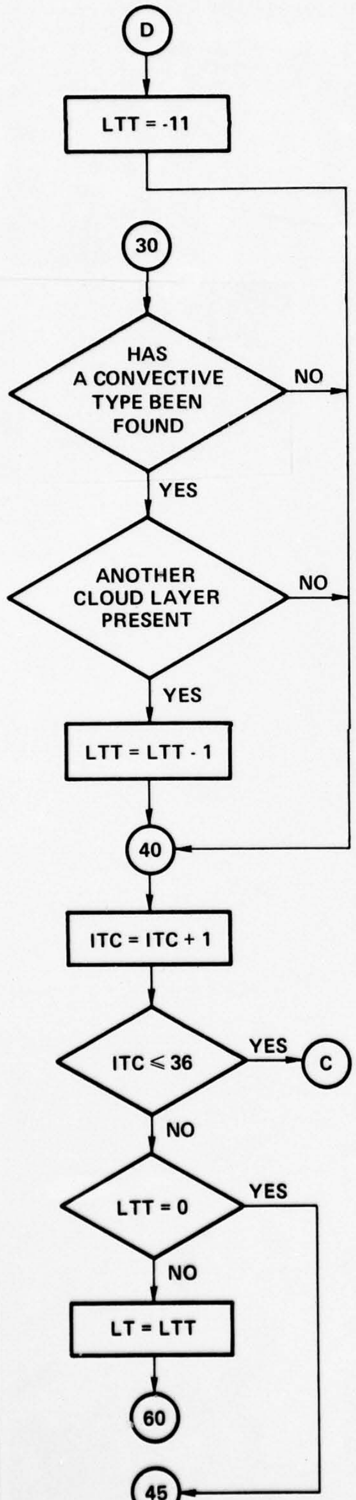
Jump to 65 if OBS/REP is not an AIRWAYS, METAR or SYNOP type.



Jump to 10 if the low cloud type word of the OBS/REP does not show a convective type to be present.

LT, the OBS/REP analysis classification is a positive two digit integer if low, middle high cloud type data is present. Ten's digit is type of low cloud. Units digit is 1 for low cloud only, 2 for low and middle or high cloud, 3 for low middle and high clouds.

Initialize intermediate classification and address of layered cloud type designator in OBS/REP.



LTT set equal to -11, if a layer with a convective type low cloud is found.

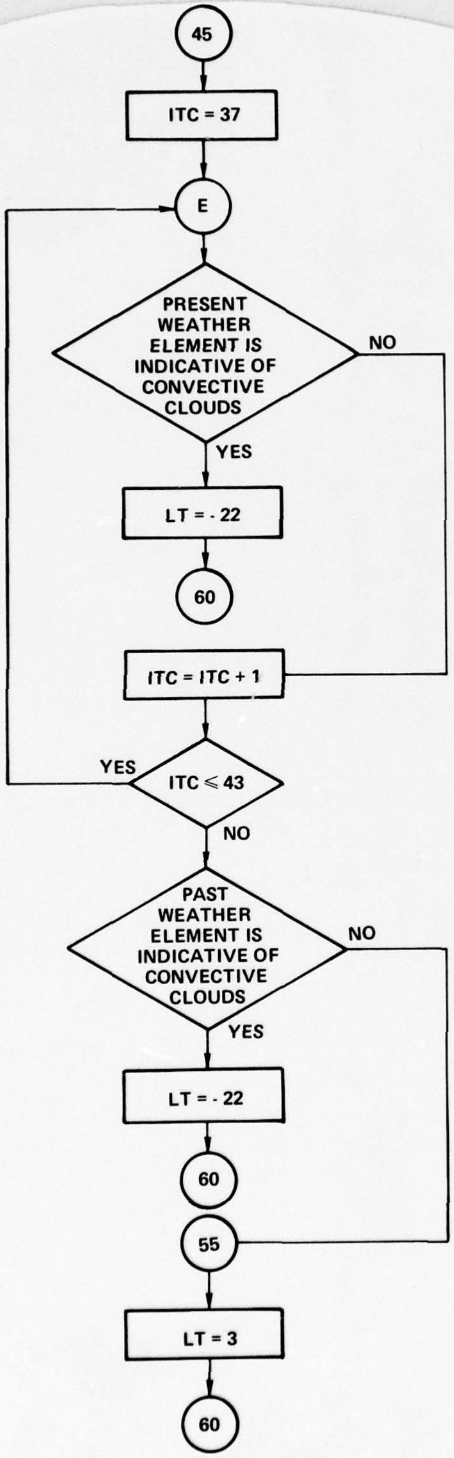
Decrement LTT by 1 for each identifiable cloud layer above the layer indicating convective type clouds which is reported.

Jump to 45 if no convective type clouds have been found in the layered cloud data.

LT is a negative integer for layered type cloud data as in AIRWAYS, METAR or the supplementary group if given in SYNOP. LT equals -11 if only convective type clouds were reported, equals -12 if convective and a layer of another type were reported, and equals -13 if a convective type plus two or more other layers of clouds were reported.

AD-A048 564 SPERRY RESEARCH CENTER SUDBURY MASS F/G 4/2
DEVELOPMENT OF CLOUD/FOG ANALYSIS AND APPLICATION SUBROUTINES F--ETC(U)
NOV 75 B R FOW, W D MOUNT DAAD07-74-C-0251
UNCLASSIFIED SCRC-CR-75-17 NL

2 OF 3
ADA048564

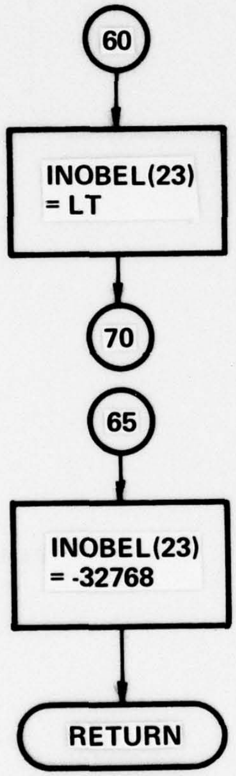


Initialize index of present weather elements in OBS/REP.

LT equals -22 if a present weather element indicative of convective clouds was found.

LT equals -22 is the past weather element is indicative of convective clouds.

LT = 3 is the classification given to a type 1, 2 or 3 OBS/REP in which the probable presence of convective clouds could not be inferred from the data.

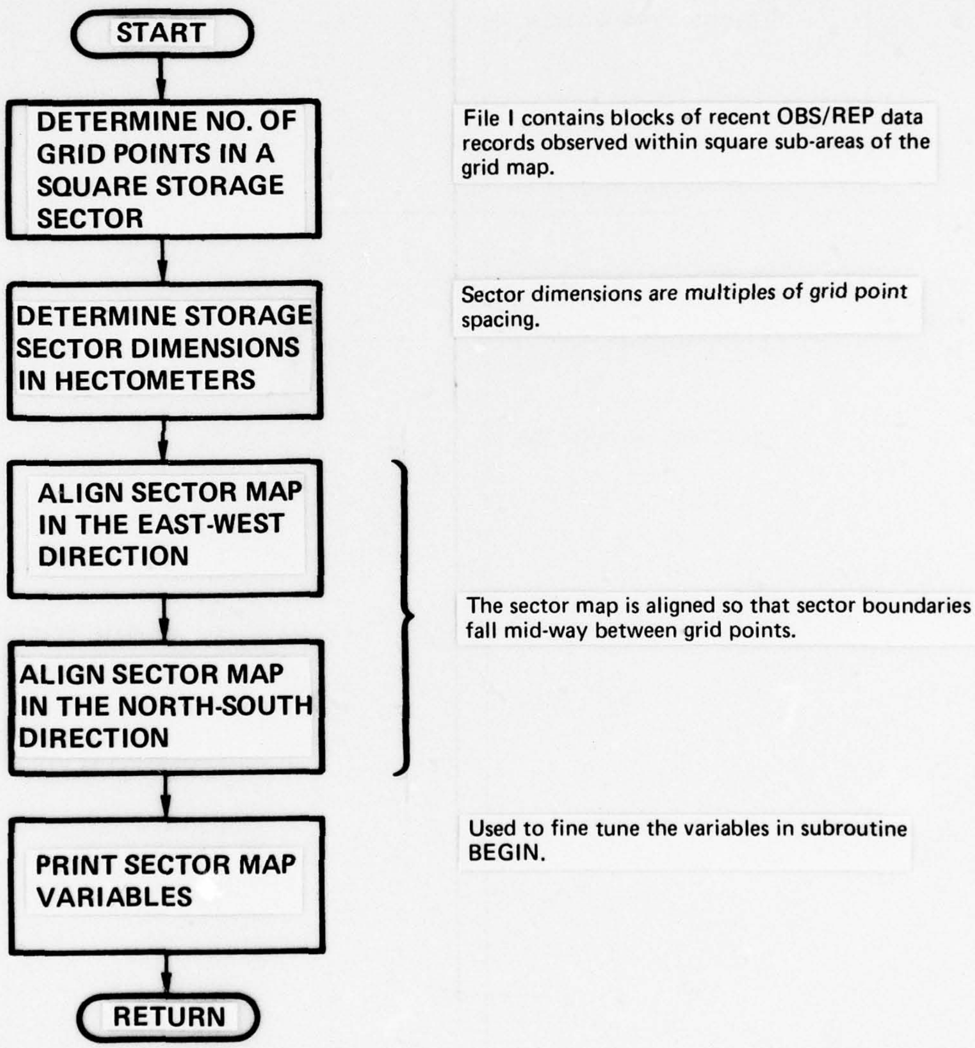


OBS/REP analysis classification word set equal to LT.

OBS/REP analysis classification word set equal to missing (i.e. -32768) for OBS/REP types other than 1, 2, or 3.

SUBROUTINE SECTOR

Establish the storage sector map for OBS/REP storage and retrieval routines. All variables used in subroutine SECTOR are defined in subroutine BEGIN.



SUBROUTINE SFDINT

Routine to interpret surface OBS/REP in terms of CFDB parameters.

Sources of input data are aviation weather reports in AIRWAYS and METAR codes and surface synoptic reports in SYNOP code.

Input Data

IX = X distance of OBS/REP site from IXREF, hectometers
IY = Y distance of OBS/REP site from IYREF, hectometers
IZ = Terrain height at OBS/REP site, meters
ITIME = Time of OBS/REP
ITYPE = Type of OBS/REP
    1 = AIRWAYS   -1 if a SPECIAL
    2 = METAR   -2 if a SPECI (SPECIAL)
    3 = SYNOP
IDD = Wind direction, 0–360 from true north
IFF = Wind speed, meters/sec
IPPP = Sea level pressure, millibars
ITT = Surface temperature, degrees Kelvin
ITD = Surface dewpoint, degrees Kelvin
ITSC = Total sky cover, 0–9   WMO code 2700
IVIS = Visibility –
    AIRWAYS – Statute miles * 10000
    METAR – Meters
    SYNOP – WMO code 4377
NWEA(J) = Present weather – from 1 to 7 elements may be input
    AIRWAYS – CFAS code 1
    METAR – WMO code 4678
    SYNOP – WMO code 4677
IPW = Past weather, 0–9   WMO code 4500
NH = Sky cover due to low or middle clouds, 0–9   WMO code 2700
ICL = Low cloud type, 0–9   WMO code 0513
IH = Height above ground of lowest cloud, 0–9   WMO code 1600
ICM = Middle cloud type, 0–9   WMO code 0515
ICH = High cloud type, 0–9   WMO code 0509
NS(J) = Sky cover due to cloud layer – from 1 to 10 layers
    AIRWAYS – CFAS code 2
    METAR – WMO code 2700
    SYNOP – WMO code 2700
ICTS(J) = Type of cloud in layer, 0–9   WMO code 0500
IHS(J) = Height of base of cloud layer
    AIRWAYS – 100's of feet
    METAR – WMO code 1677
    SYNOP – WMO code 1677
ITHN(J) = Cloud layer thickness indicator
    1 if thin
    Missing if not thin

ICLG = Ceiling designator – first two digits are the index No. J of the ceiling layer. Third digit has a following meaning

1 = Measured
2 = Aircraft
3 = Balloon
4 = Radar
5 = Estimated
6 = Indefinite

ICLGV = Characteristic of ceiling

Missing = Not variable
1 = Variable

IVISC = Visibility characteristics

Missing = Not variable
1 = Variable

Cloud/fog data base parameters

IVALU = Information value of the OBS/REP (1–10)

0 indicates no data useable for determining any CFDB params.
10 indicates an OBS/REP with all needed data present and useable.
1 to 9 indicates an OBS/REP with some missing or non-useable data.

NTCLC = Total cloud cover. (00 – 100)
NCEIL = Height of ceiling layer (AGL), dekameters + type of ceiling digit as per third digit of ICLG. Minus if variable.
MINBAS = Height of base of lowest cloud (AGL), dekameters.
MAXTOP = Height of the top of highest cloud (AGL), dekameters.
MSPWE = Most significant present weather element (WMO code 4677)
NVV = Prevailing visibility at surface, meters. Negative if variable.
LCOV(9) = Percent cloud cover in the CFDB layers

Derived layered cloud information

NUMLAY = Number of layers generated
KIND = Kind of cloud layer

1 = Low
2 = Middle
3 = High
4 = Fog
5 = Lowest cloud
6 = Clear layer

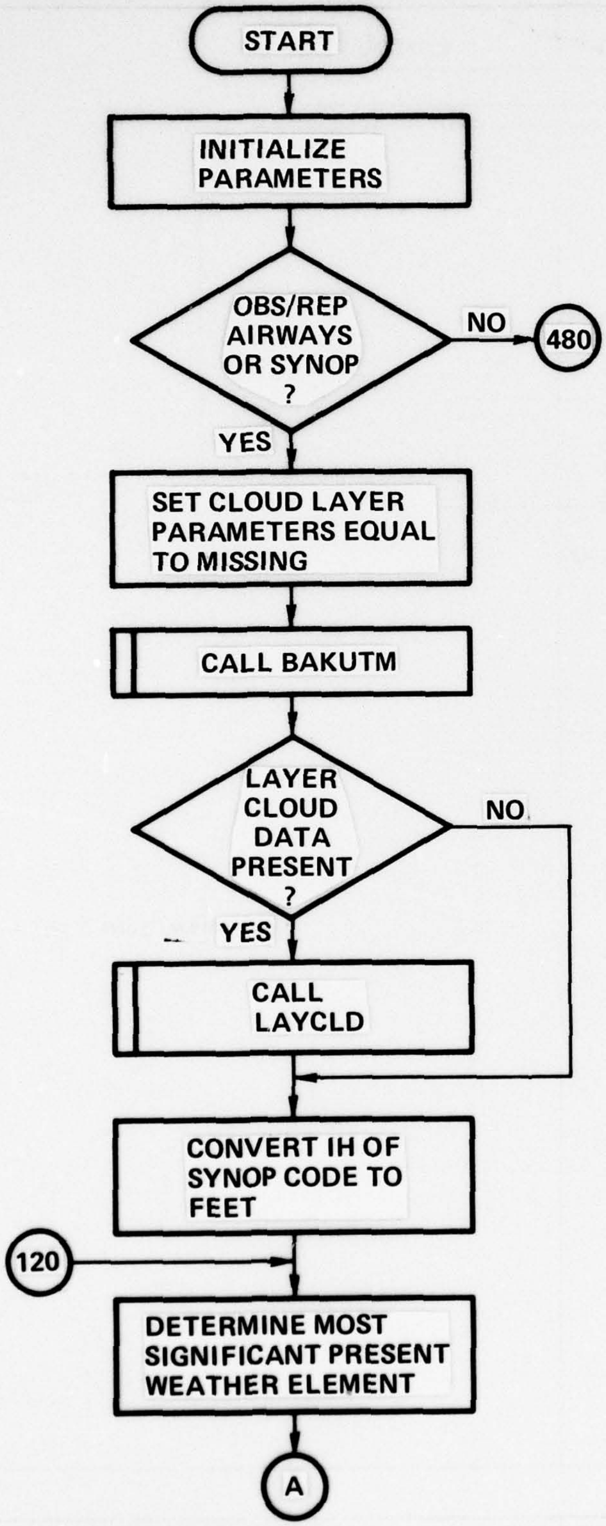
ITHIN = Thin layer designator

MISSING = Not thin
1 = Thin

COVER = Cloud cover in layer (0.0 – 1.0)
BASE = Height of the base of layer, feet.
TOP = Height of top of cloud layer, feet.

Map and window data

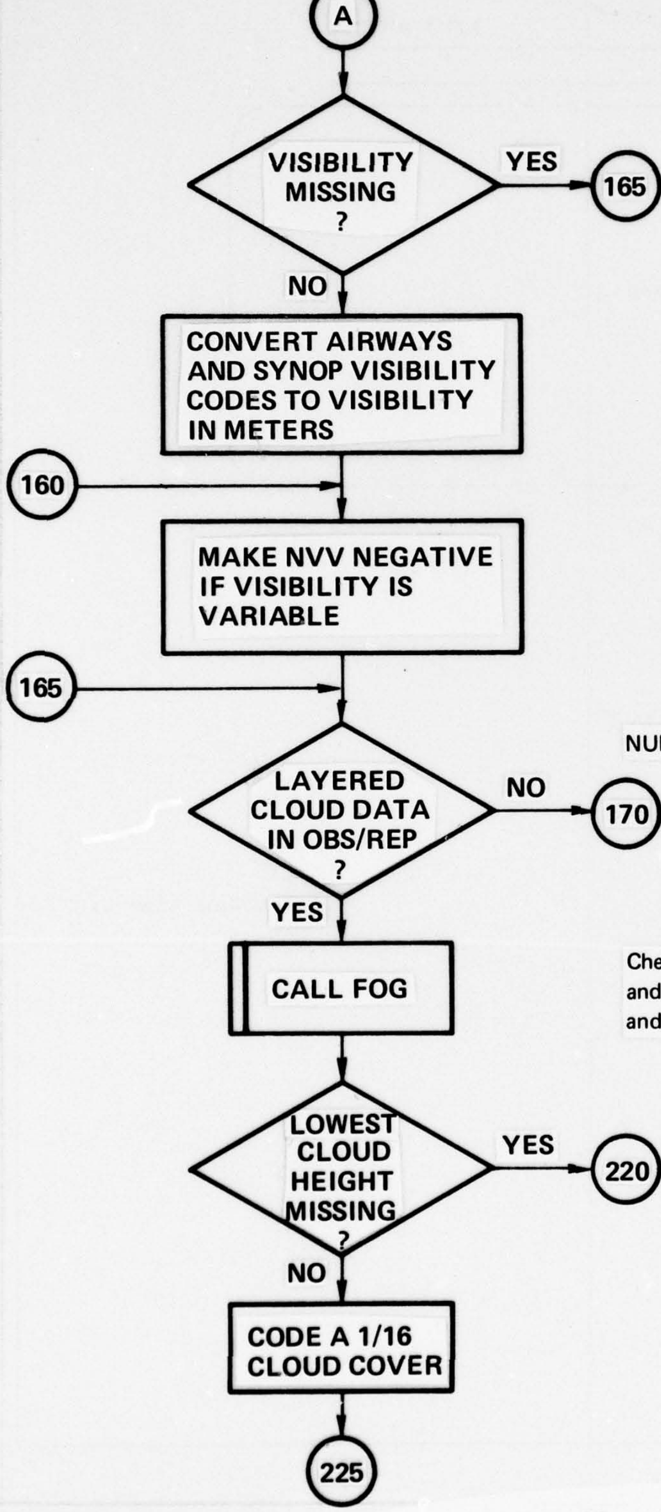
XREF = East-west UTM grid coordinate of lower left hand corner of the window, KM.
YREF = North-south UTM grid coordinate of lower left hand corner of the window, KM.
CMRD = Central meridian of window



Calculate latitude of OBS/REP.

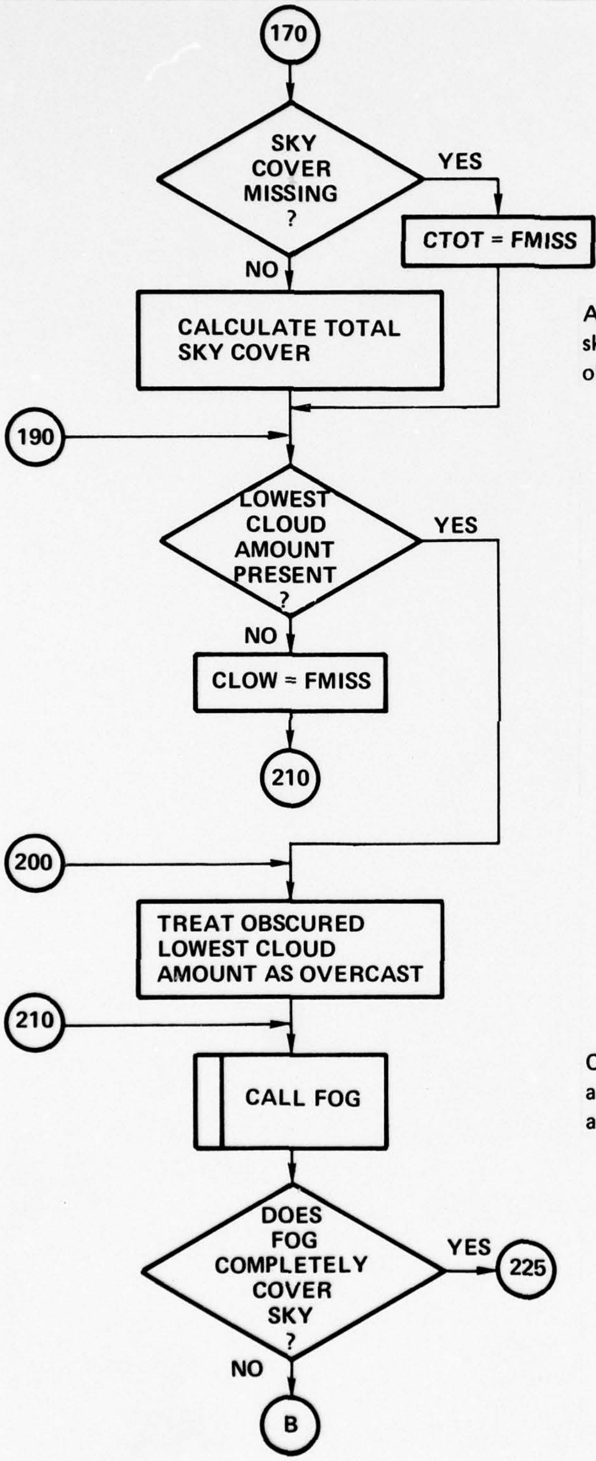
Construct cloud layers.

Convert IH WMO code 1600 to feet. If IH = M or >8, set lowest cloud base, HITLOW = M.



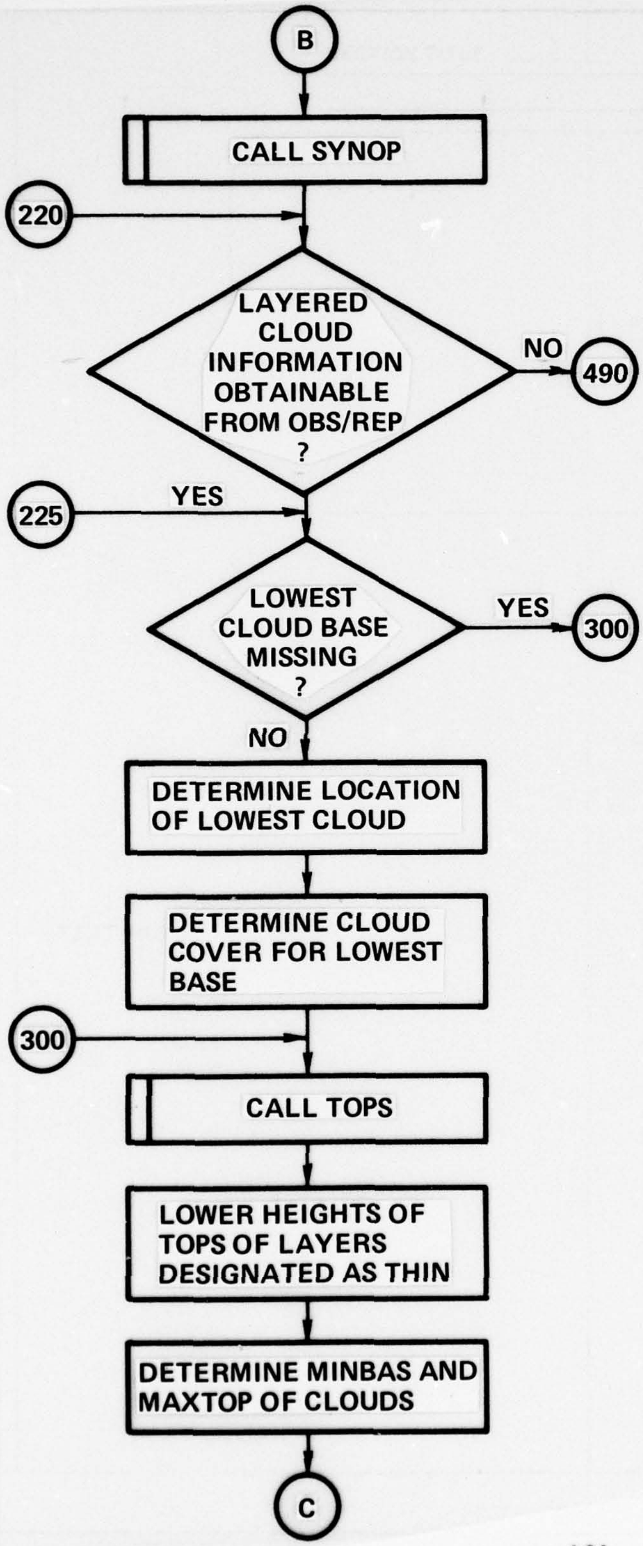
NUMLAY = No. of cloud layers constructed thus far.

Checks for fog and estimates percentage cloud cover and tops of clouds on basis of horizontal visibility and type of fog.



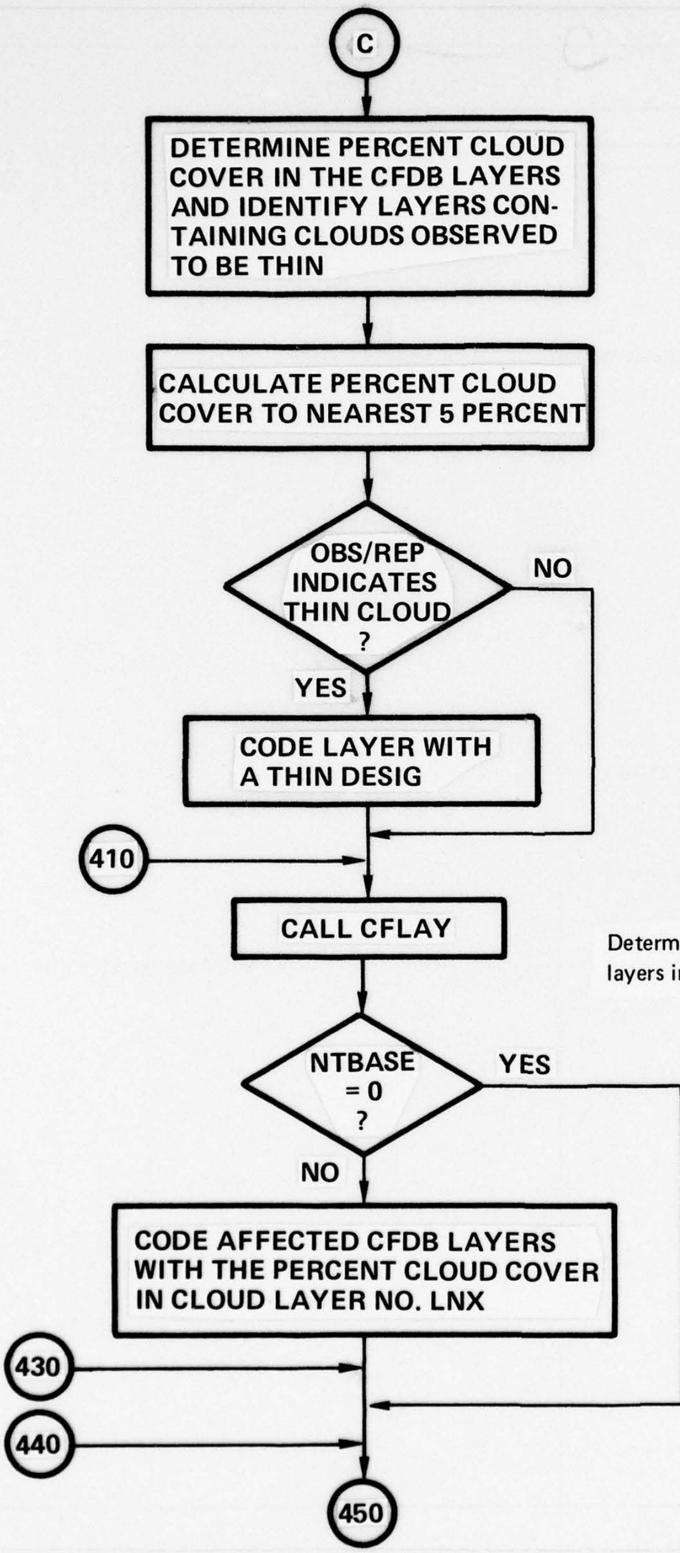
Assure low-middle cloud cover not greater than total sky cover when total sky cover not missing or obscured.

Check for fog and estimate percentage cloud cover and tops of cloud layers from horizontal visibility and type of fog.



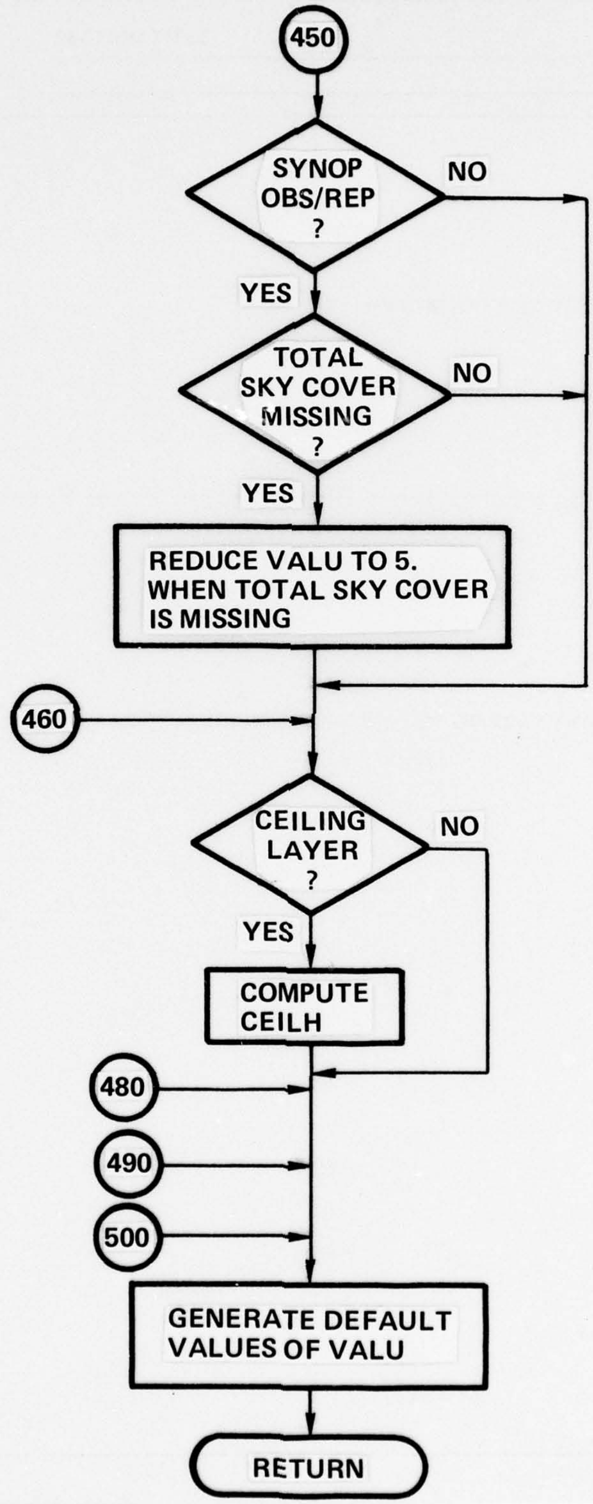
Construct cloud layers from mandatory SYNOP type data.

Determine cloud tops.



Determine index nos. of lowest and highest CFDB layers influenced by cloud layer no.

C ← cap C

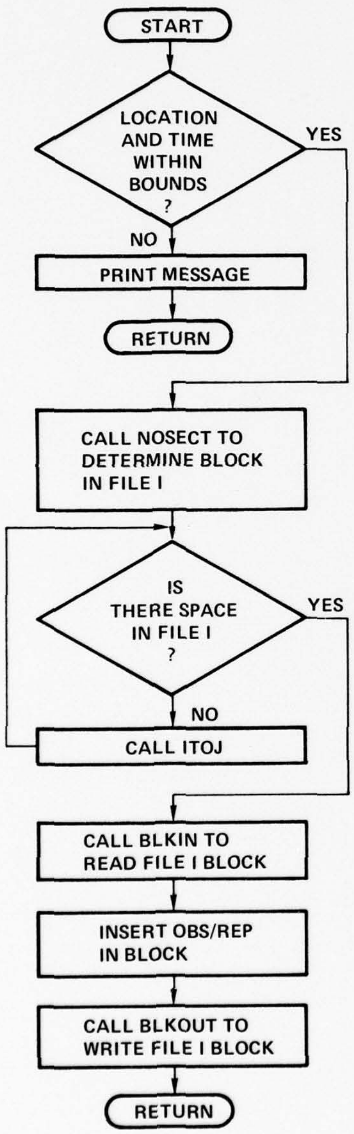

450
SYNOP OBS/REP ?
NO
YES
TOTAL SKY COVER MISSING ?
NO
YES
REDUCE VALU TO 5. WHEN TOTAL SKY COVER IS MISSING
460
CEILING LAYER ?
NO
YES
COMPUTE CEILH
480
490
500
GENERATE DEFAULT VALUES OF VALU
RETURN

SUBROUTINE STOREC (IREC)

Stores an OBS/REP in the OBS/REP data base.

IREC = Starting address of OBS/REP from calling routine.

Input — IREC = Starting address of OBS/REP.



If the location of the OBS/REP is outside the boundary of the sector map, the following message is printed — "DATA RECORD RECEIVED WAS TOO DISTANT FOR STORAGE".

If the observation time of the OBS/REP indicates old data the following message is printed — "DATA RECORD RECEIVED TOO LATE FOR STORAGE".

Sector numbers correspond to block numbers in file I.

To store a new OBS/REP there must be space in the core array ITABLE and in the file I block.

Generate a new file J block containing the oldest OBS/REP's in file I.

Mass storage to core transfer.

OBS/REP's within a block are sorted on observation time.

Core to mass storage transfer.

SUBROUTINE SYNOP (CTOT, CLOW, HLOW, LOWT, MIDT, NHIT, NWEA, DLAT, VAL, MSPW)

Routine to convert total cloud cover, lowest cloud cover, lowest base, and cloud types into layered cloud information.

Inputs

CTOT = Total cloud cover (range 0 - 1)
CLOW = Lowest cloud cover (range 0 - 1)
HLOW = Lowest cloud base in feet
LOWT = Low cloud type
MIDT = Middle cloud type
NHIT = High cloud type
NWEA = Present weather
DLAT = Latitude

Outputs

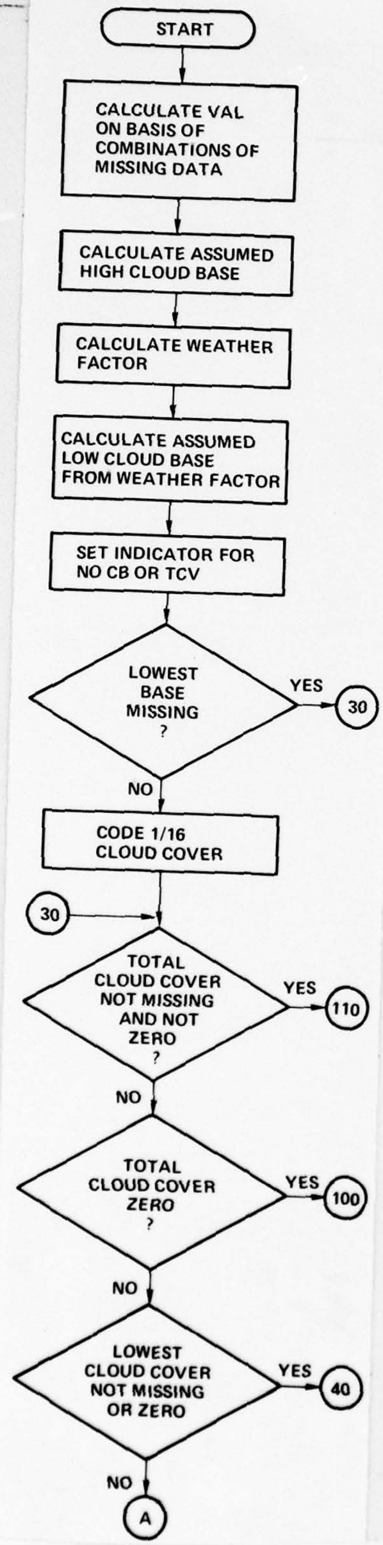
VAL = Indicator for combinations of missing data (0.0 - 10.0)
MSPW = Most significant present weather category

Derived layered cloud information on COMMON/CLOUDS/

NUMLAY = Number of layers generated (initialized before calling SYNOP)
KIND = Kind of cloud layer
- 1 = Low
- 2 = Middle
- 3 = High
- 4 = Fog
- 5 = Lowest cloud
- 6 = Clear layer

ITHIN = Thin layer designator
MISSING = Not thin
- 1 = Thin

COVER = Cloud cover in layer (0.0 - 1.0)
BASE = Height of the base of layer, feet.
TOP = Height of top of cloud layer, feet.



Enter with no. of layers (NUMLAY) initialized.

Using CTOT, CLOW, LOWT, MIDT, HLOW, and NHIT.

BASHI = high cloud base

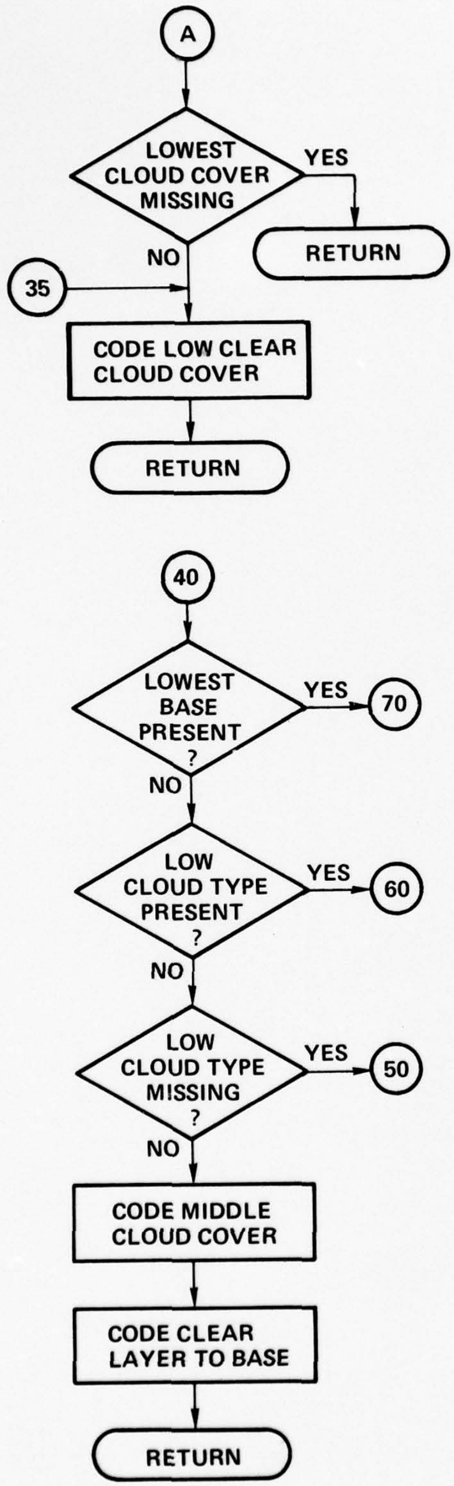
NCB

Bump NUMLAY and set NUMLAY'th values in COMMON /CLOUDS/

Check CTOT

Check CTOT

Check CLOW



Check CLOW

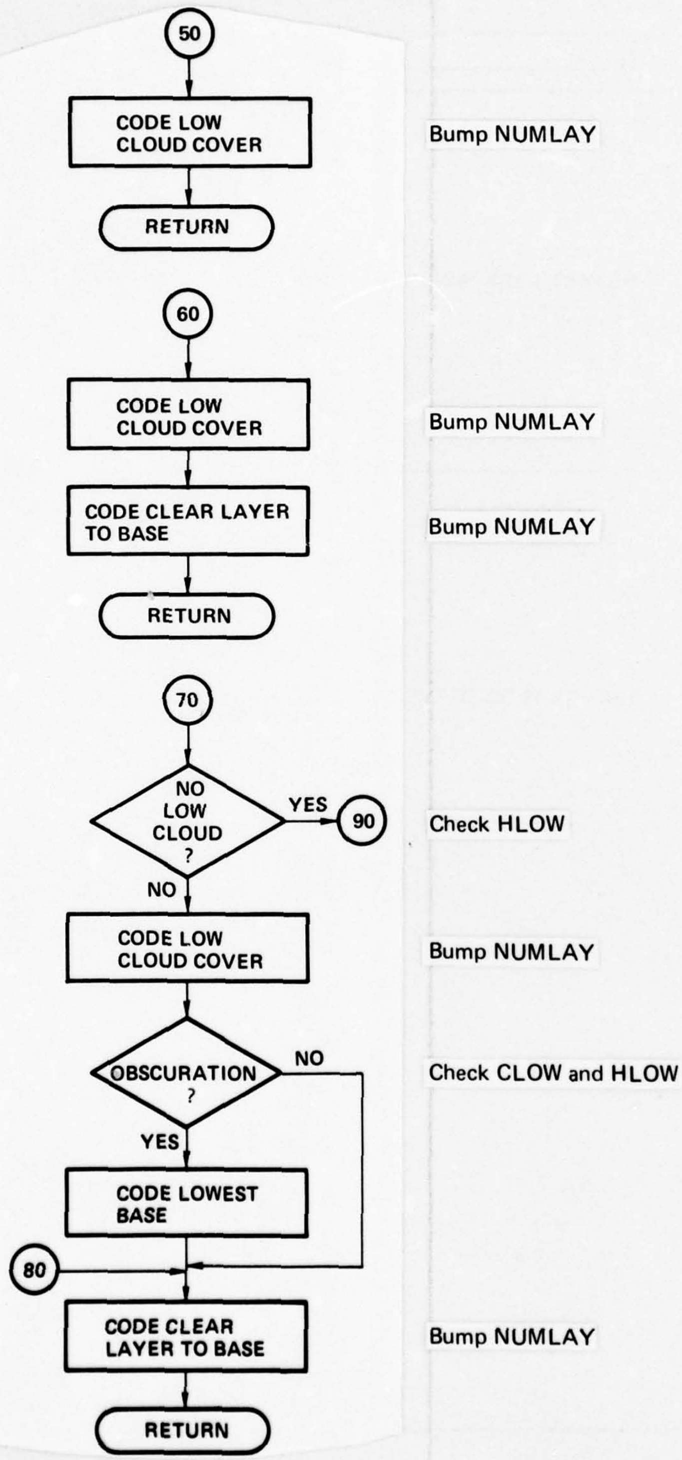
Bump NUMLAY

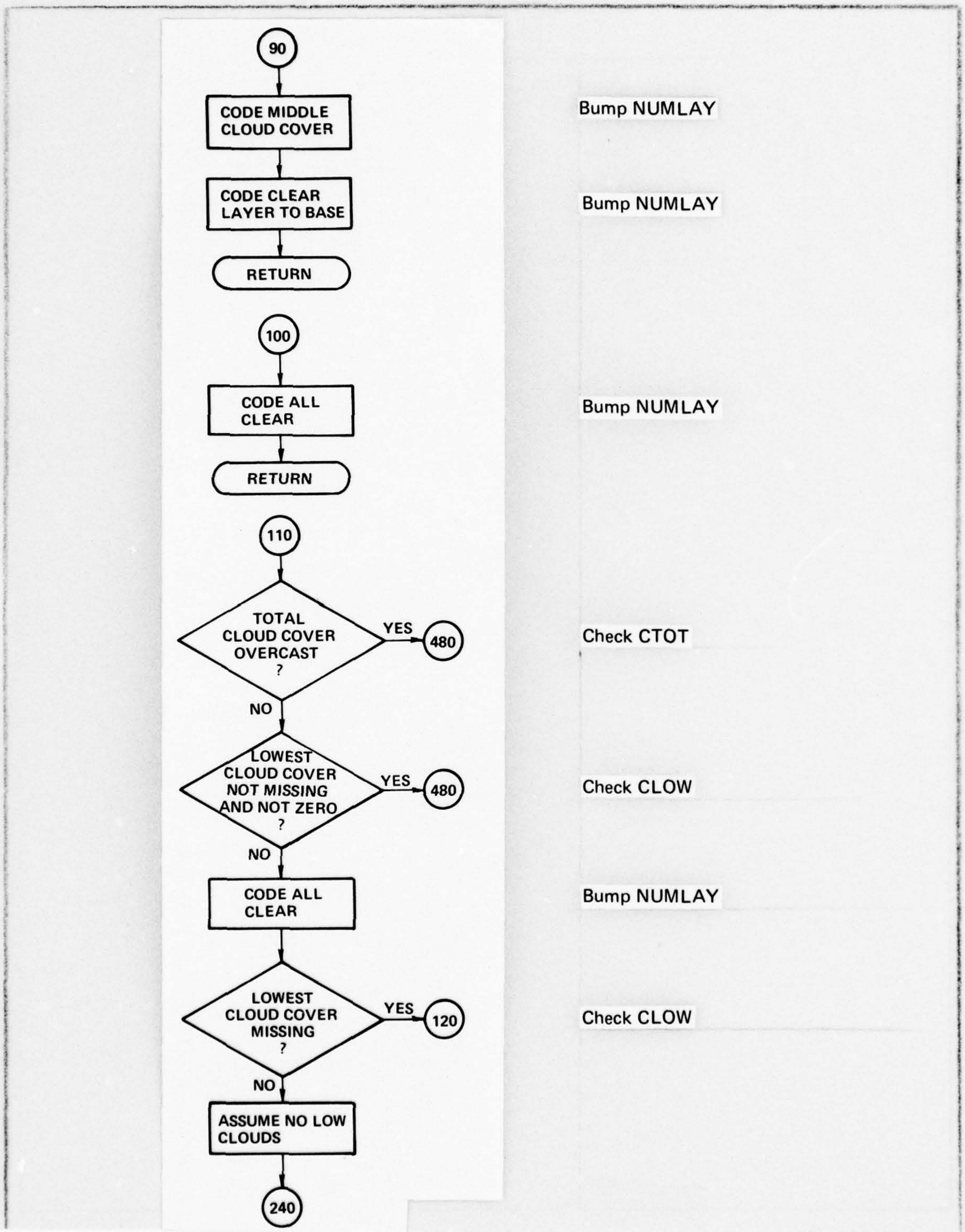
Check HLOW

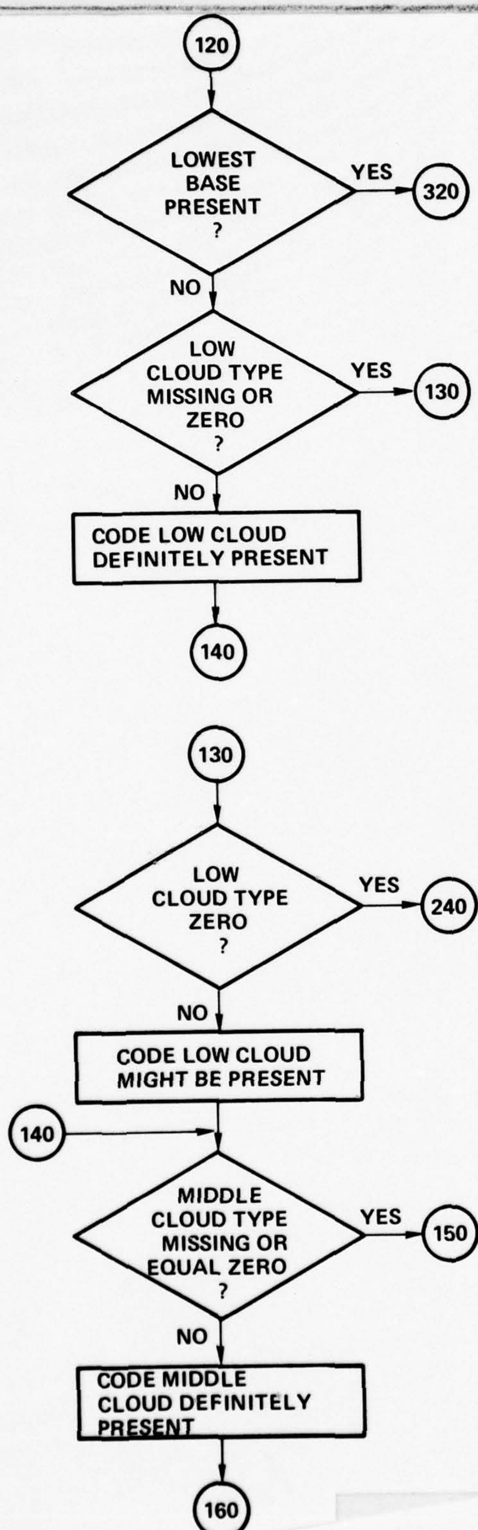
Check LOWT

Bump NUMLAY

Bump NUMLAY


50
CODE LOW CLOUD COVER
Bump NUMLAY
RETURN
60
CODE LOW CLOUD COVER
Bump NUMLAY
CODE CLEAR LAYER TO BASE
Bump NUMLAY
RETURN
70
NO LOW CLOUD ?
YES
90
Check HLOW
NO
CODE LOW CLOUD COVER
Bump NUMLAY
OBSCURATION ?
NO
Check CLOW and HLOW
YES
CODE LOWEST BASE
80
CODE CLEAR LAYER TO BASE
Bump NUMLAY
RETURN


90
CODE MIDDLE CLOUD COVER
Bump NUMLAY
CODE CLEAR LAYER TO BASE
Bump NUMLAY
RETURN
100
CODE ALL CLEAR
Bump NUMLAY
RETURN
110
TOTAL CLOUD COVER OVERCAST ?
YES
480
NO
Check CTOT
LOWEST CLOUD COVER NOT MISSING AND NOT ZERO ?
YES
480
NO
Check CLOW
CODE ALL CLEAR
Bump NUMLAY
LOWEST CLOUD COVER MISSING ?
YES
120
NO
Check CLOW
ASSUME NO LOW CLOUDS
240

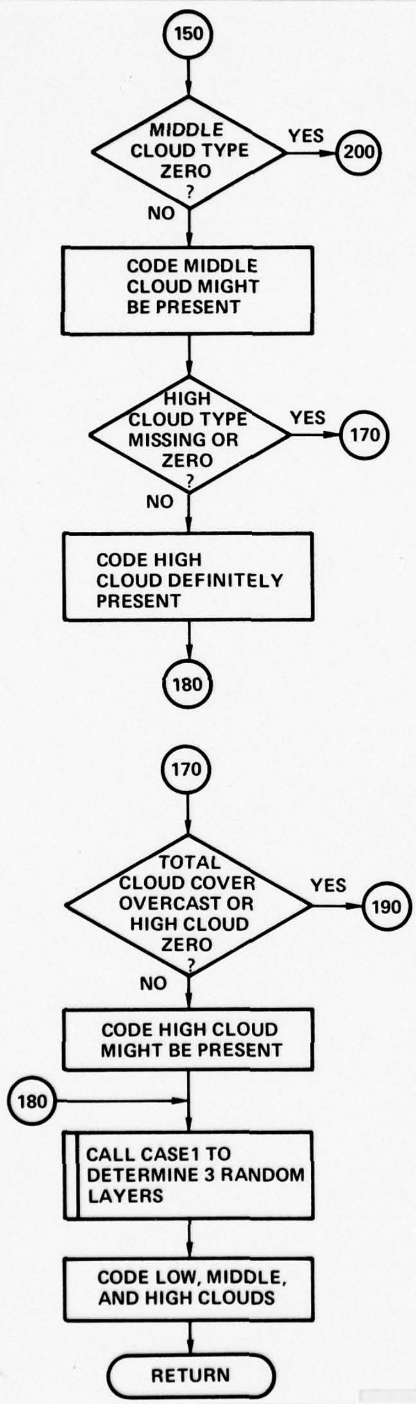


Check HLOW

Check LOWT

Check LOWT

Check MIDT

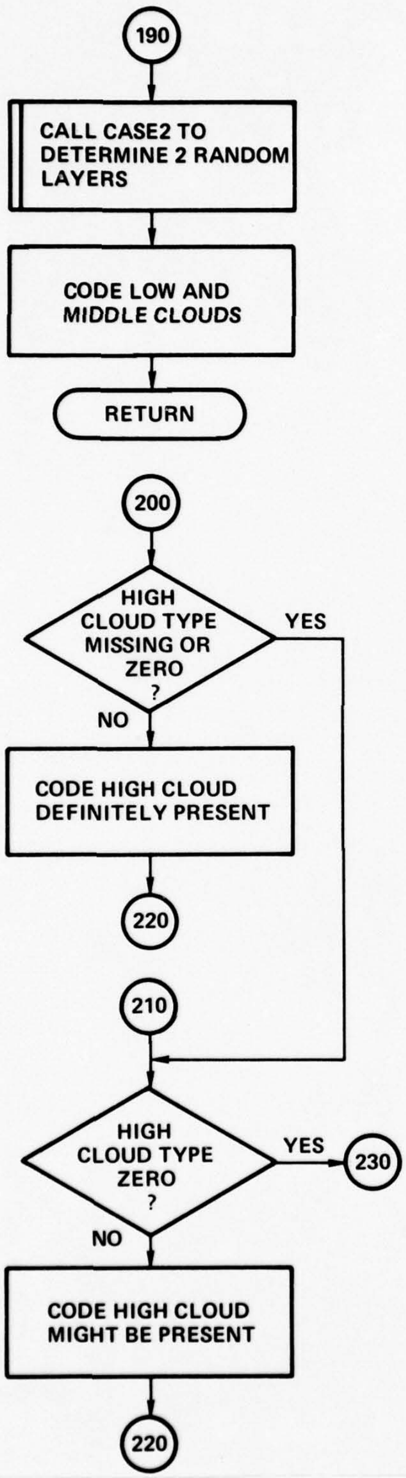


Check MIDT

NHIT

Check CTOT and NHIT

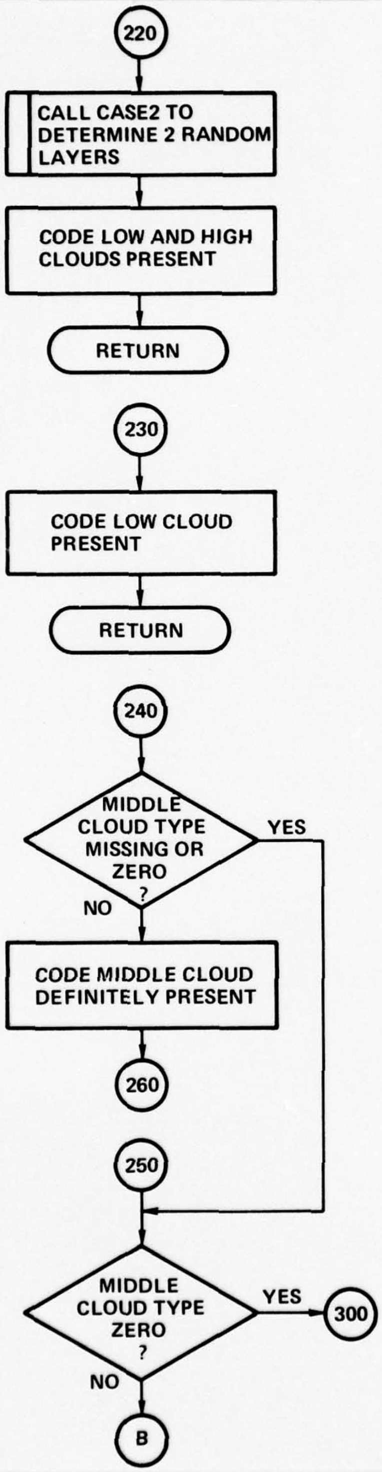
Bump NUMLAY for each



Bump NUMLAY for each

Check NHIT

Check NHIT

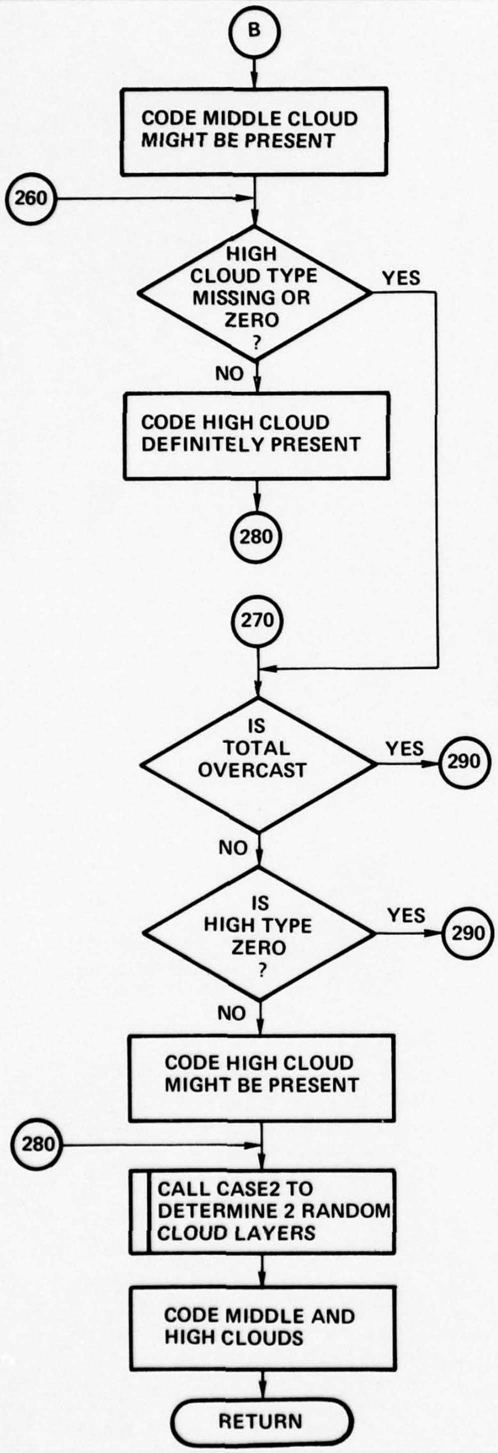


Bump NUMLAY for each

Bump NUMLAY

Check MIDT

Check MIDT

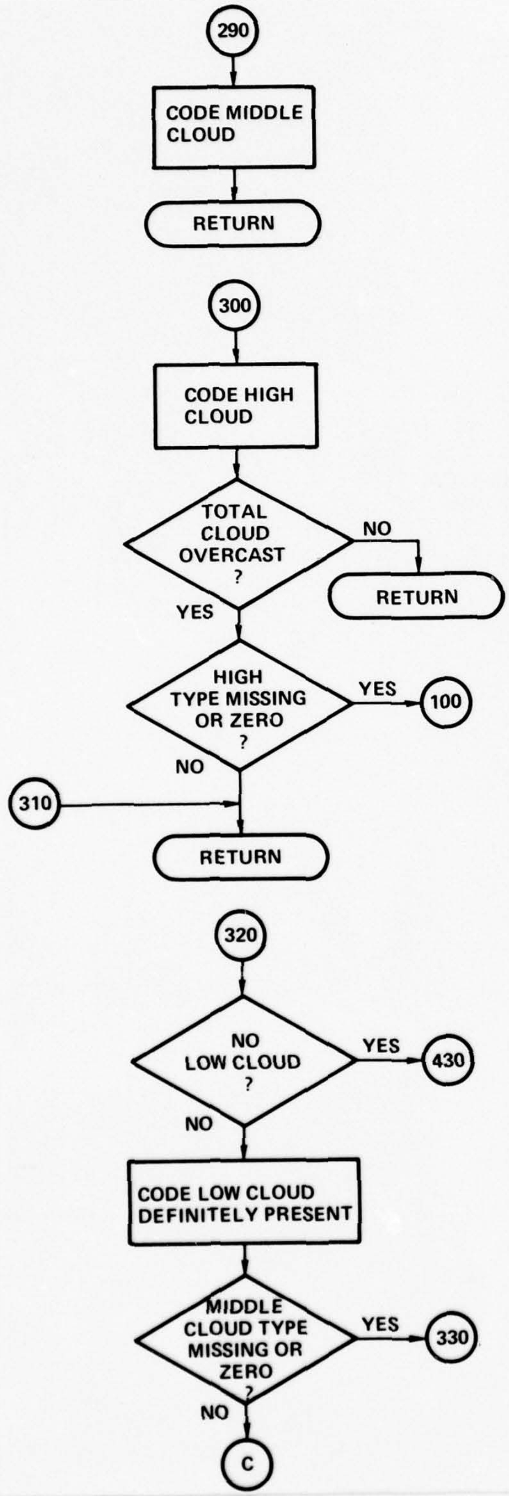


Check NHIT

Check CTOT

Check NHIT

Bump NUMLAY for each



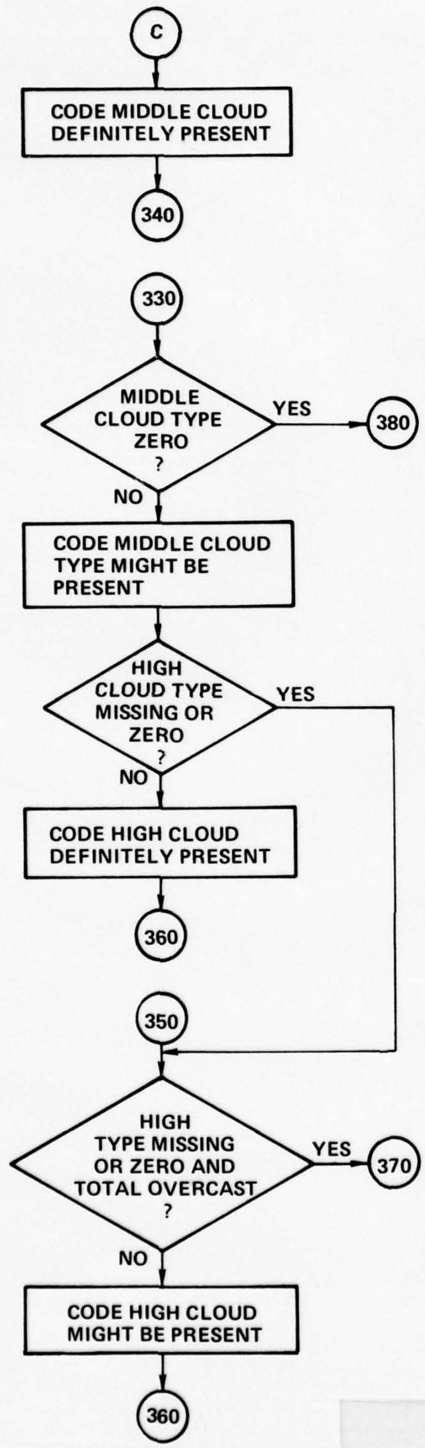
Bump NUMLAY

Check CTOT

Check NHIT

Check HLOW

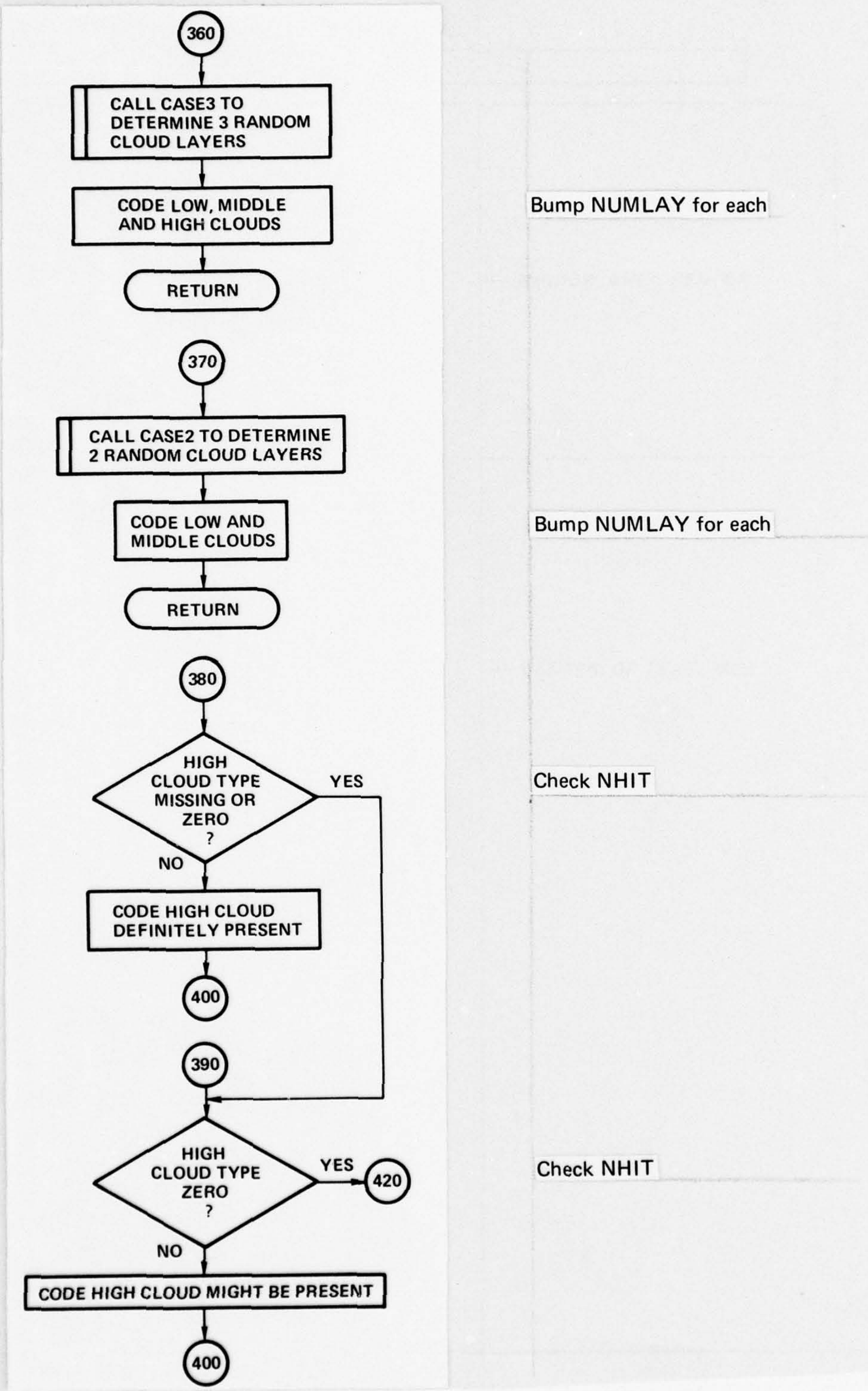
Check MIDT

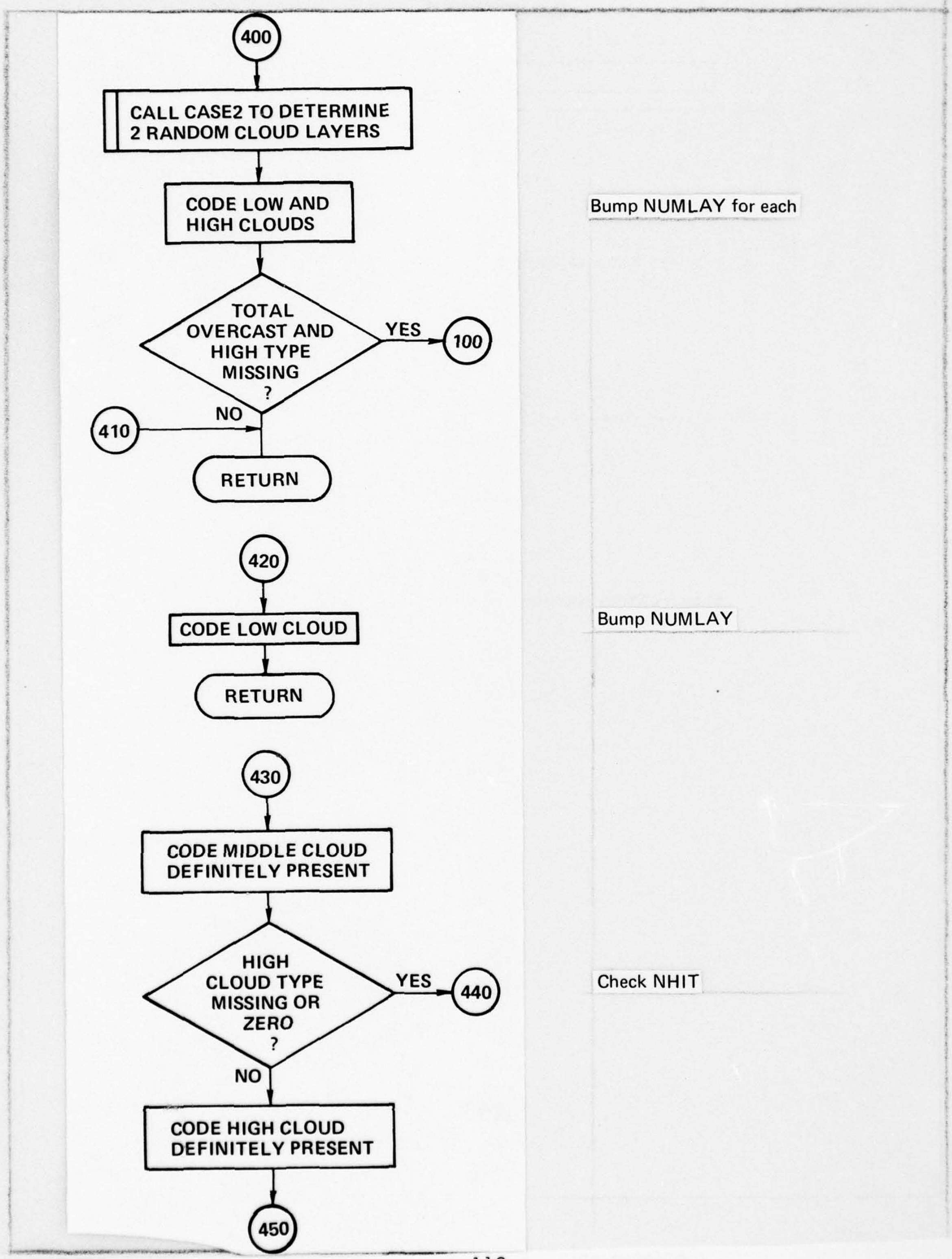


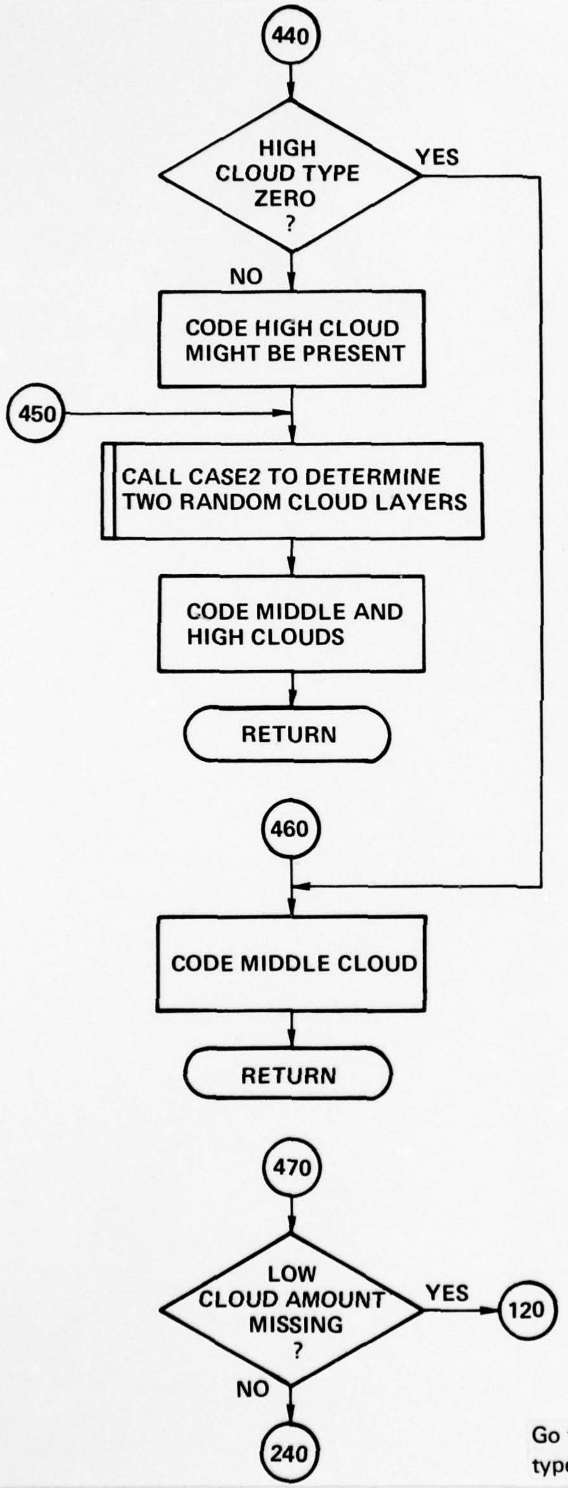
Check MIDT

Check NHIT

Check CTOT and NHIT


360
CALL CASE3 TO DETERMINE 3 RANDOM CLOUD LAYERS
CODE LOW, MIDDLE AND HIGH CLOUDS
Bump NUMLAY for each
RETURN
370
CALL CASE2 TO DETERMINE 2 RANDOM CLOUD LAYERS
CODE LOW AND MIDDLE CLOUDS
Bump NUMLAY for each
RETURN
380
HIGH CLOUD TYPE MISSING OR ZERO ?
YES
NO
Check NHIT
CODE HIGH CLOUD DEFINITELY PRESENT
400
390
HIGH CLOUD TYPE ZERO ?
YES
420
NO
Check NHIT
CODE HIGH CLOUD MIGHT BE PRESENT
400

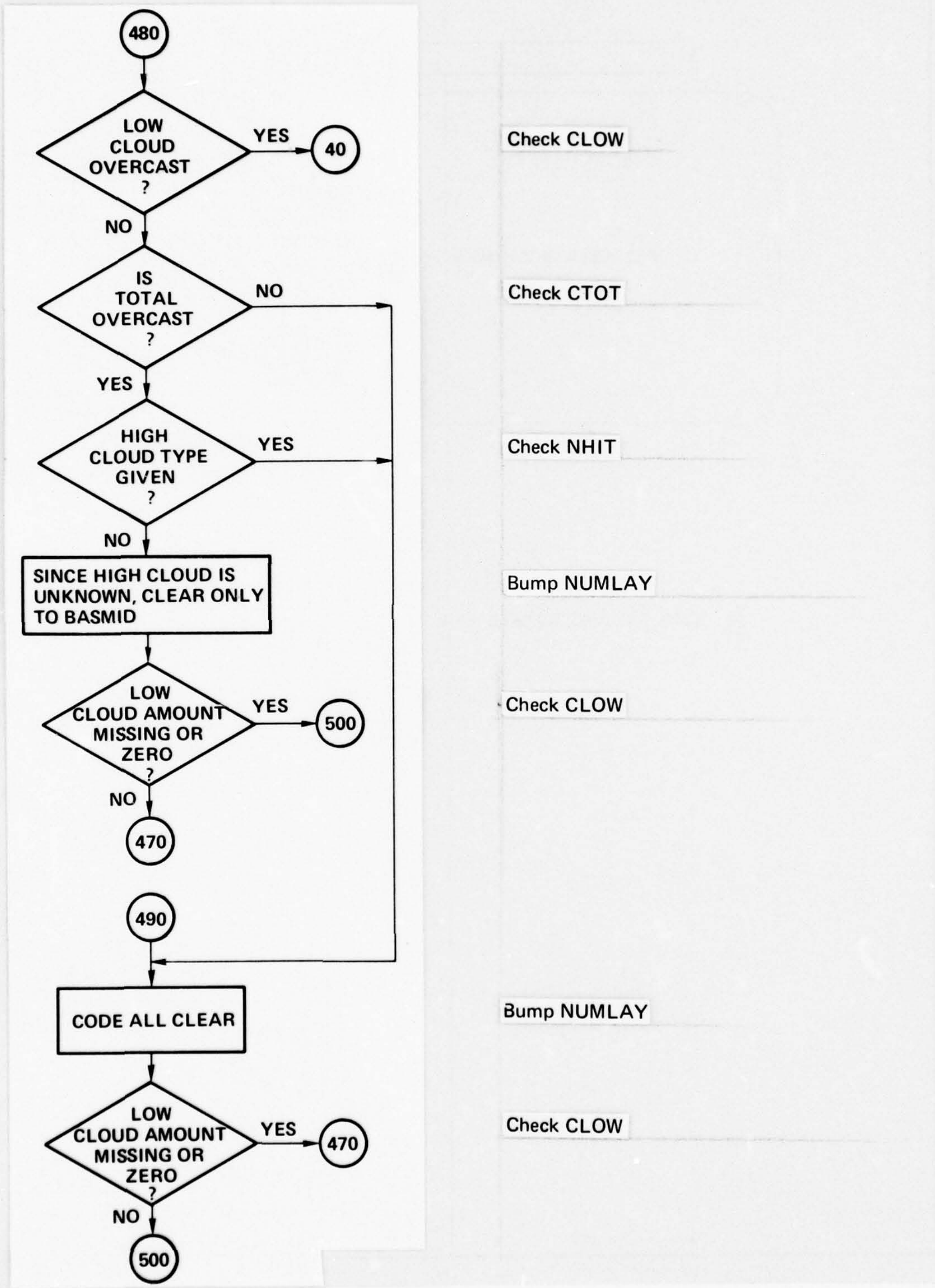

400
CALL CASE2 TO DETERMINE 2 RANDOM CLOUD LAYERS
CODE LOW AND HIGH CLOUDS
TOTAL OVERCAST AND HIGH TYPE MISSING ?
YES
100
NO
410
RETURN
420
CODE LOW CLOUD
RETURN
430
CODE MIDDLE CLOUD DEFINITELY PRESENT
HIGH CLOUD TYPE MISSING OR ZERO ?
YES
440
NO
CODE HIGH CLOUD DEFINITELY PRESENT
450
Bump NUMLAY for each
Bump NUMLAY
Check NHIT

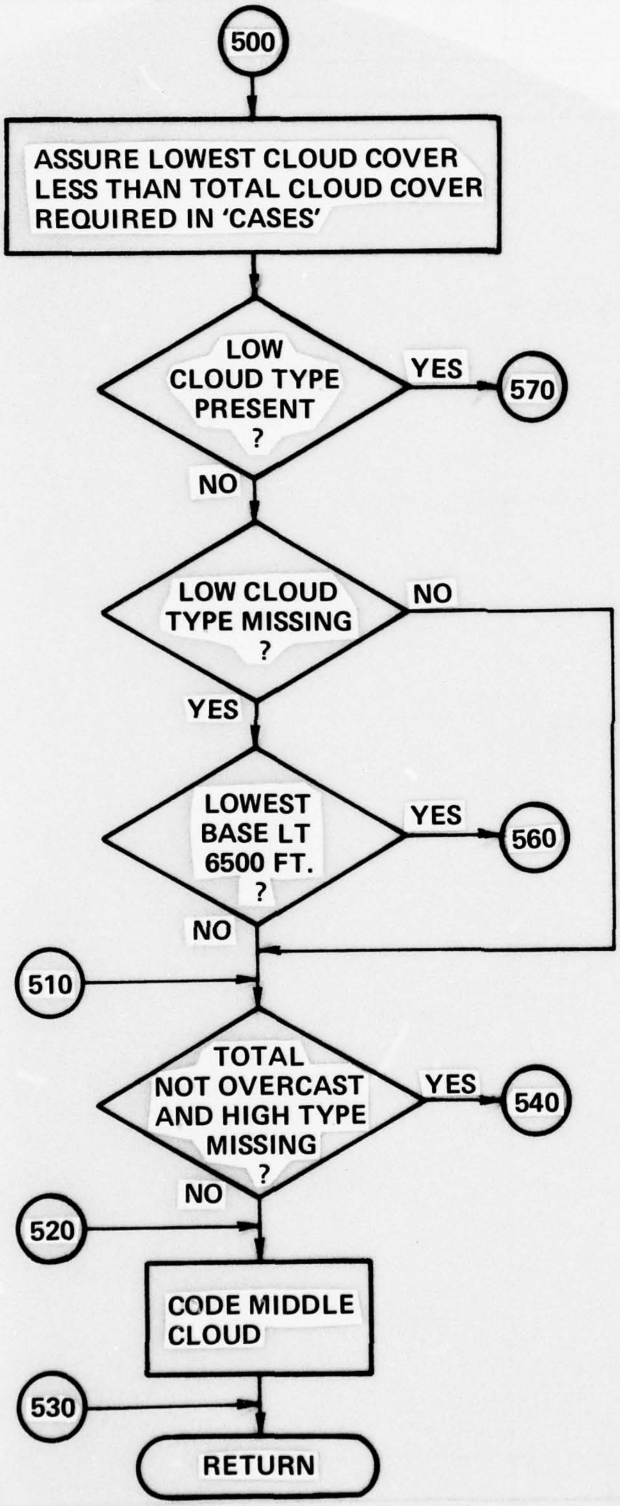


Check NHIT

Bump NUMLAY for each

Go to 240 if no low clouds to test middle and high types and total cloud cover.


480
LOW CLOUD OVERCAST ?
YES
40
NO
IS TOTAL OVERCAST ?
NO
YES
HIGH CLOUD TYPE GIVEN ?
YES
NO
SINCE HIGH CLOUD IS UNKNOWN, CLEAR ONLY TO BASMID
LOW CLOUD AMOUNT MISSING OR ZERO ?
YES
500
NO
470
490
CODE ALL CLEAR
LOW CLOUD AMOUNT MISSING OR ZERO ?
YES
470
NO
500
Check CLOW
Check CTOT
Check NHIT
Bump NUMLAY
Check CLOW
Bump NUMLAY
Check CLOW



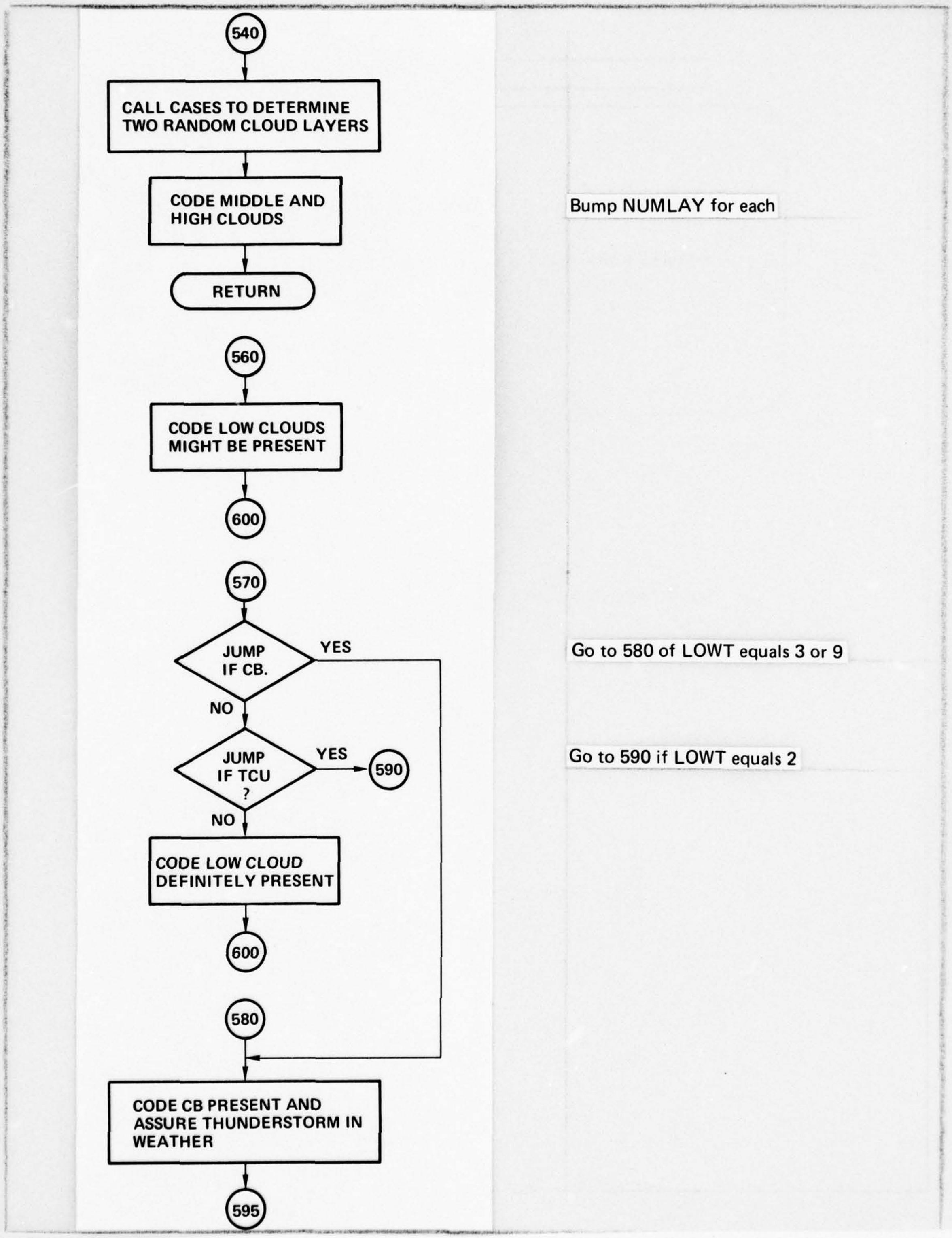
Check LOWT

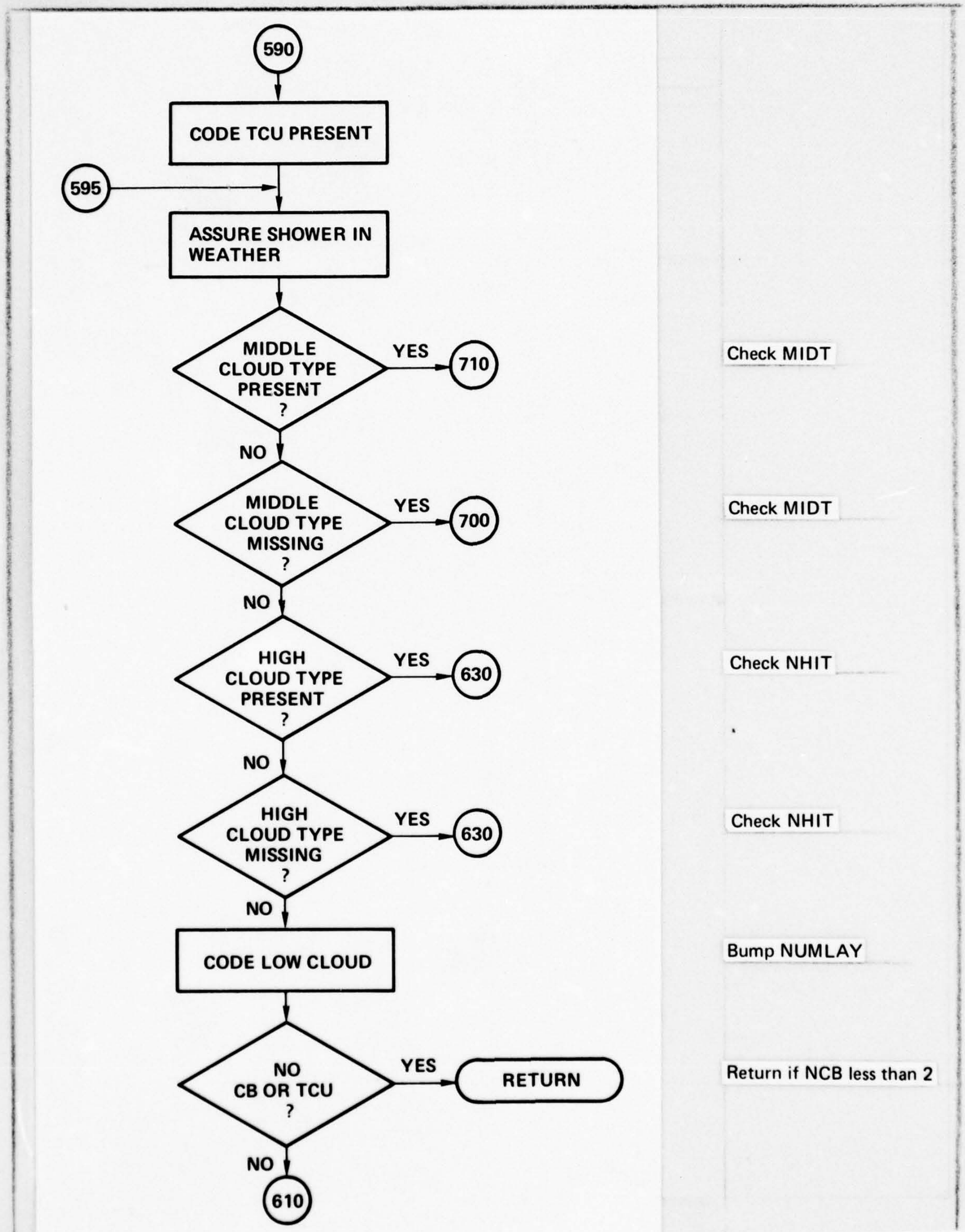
Check LOWT

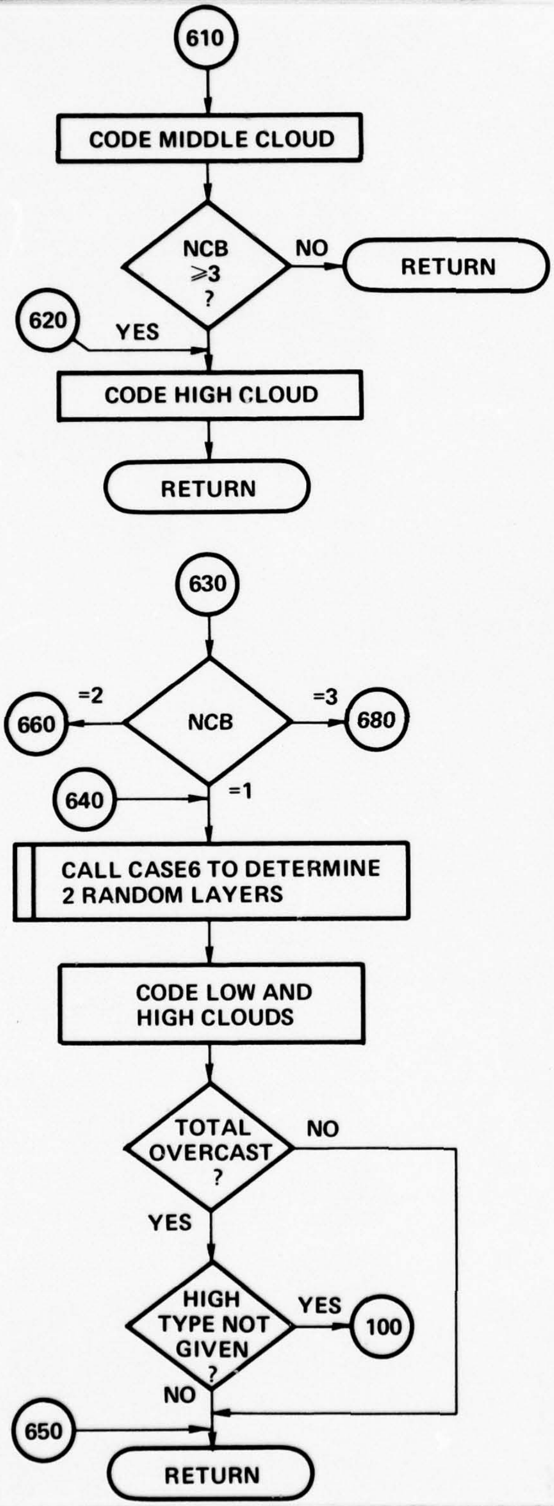
Check HLOW

Check NHIT and CTOT

Bump NUMLAY


540
CALL CASES TO DETERMINE TWO RANDOM CLOUD LAYERS
CODE MIDDLE AND HIGH CLOUDS
Bump NUMLAY for each
RETURN
560
CODE LOW CLOUDS MIGHT BE PRESENT
600
570
JUMP IF CB.
YES
NO
Go to 580 of LOWT equals 3 or 9
JUMP IF TCU ?
YES
590
NO
Go to 590 if LOWT equals 2
CODE LOW CLOUD DEFINITELY PRESENT
600
580
CODE CB PRESENT AND ASSURE THUNDERSTORM IN WEATHER
595


590
CODE TCU PRESENT
595
ASSURE SHOWER IN WEATHER
MIDDLE CLOUD TYPE PRESENT ?
YES
710
NO
MIDDLE CLOUD TYPE MISSING ?
YES
700
NO
HIGH CLOUD TYPE PRESENT ?
YES
630
NO
HIGH CLOUD TYPE MISSING ?
YES
630
NO
CODE LOW CLOUD
NO CB OR TCU ?
YES
RETURN
NO
610
Check MIDT
Check MIDT
Check NHIT
Check NHIT
Bump NUMLAY
Return if NCB less than 2

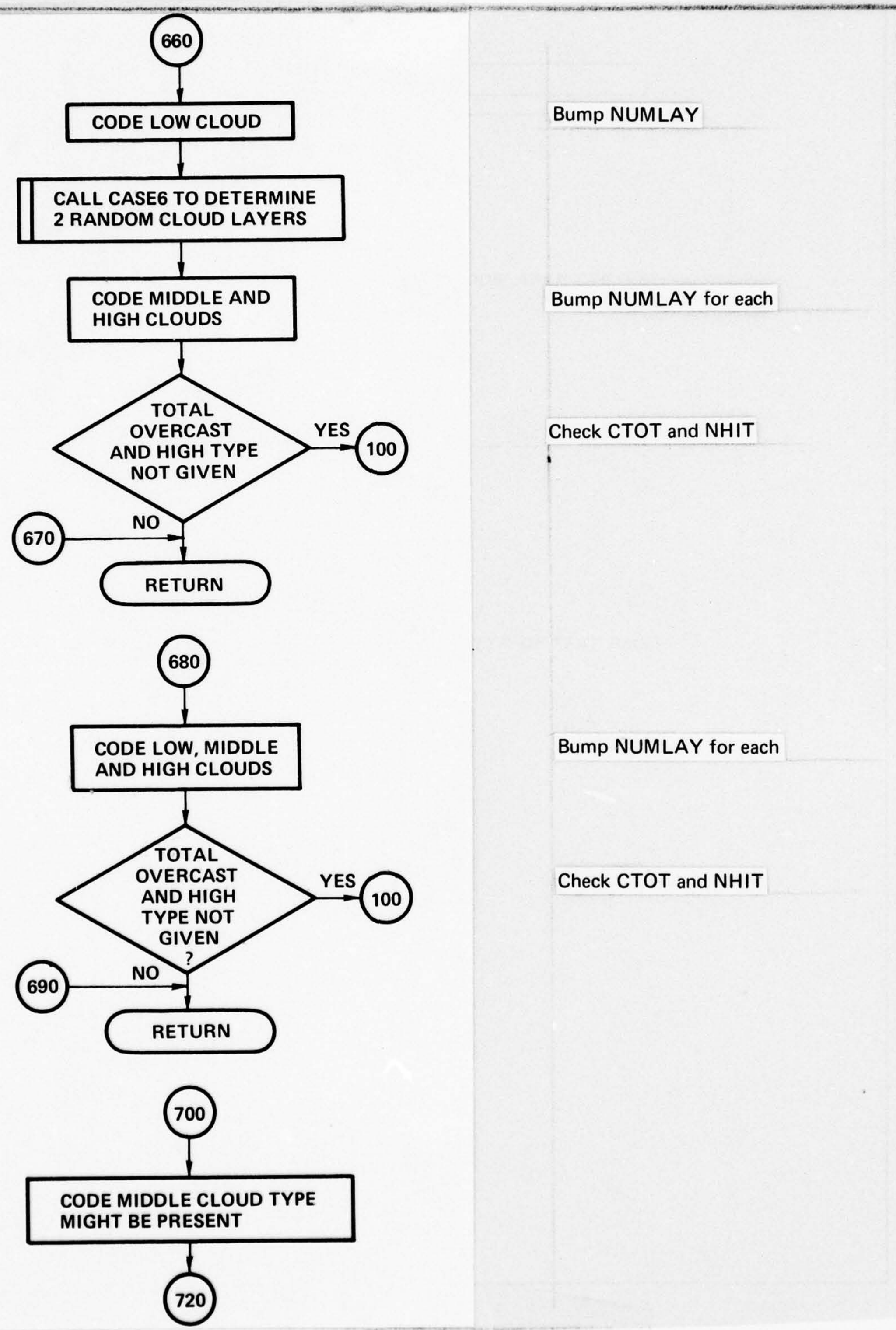


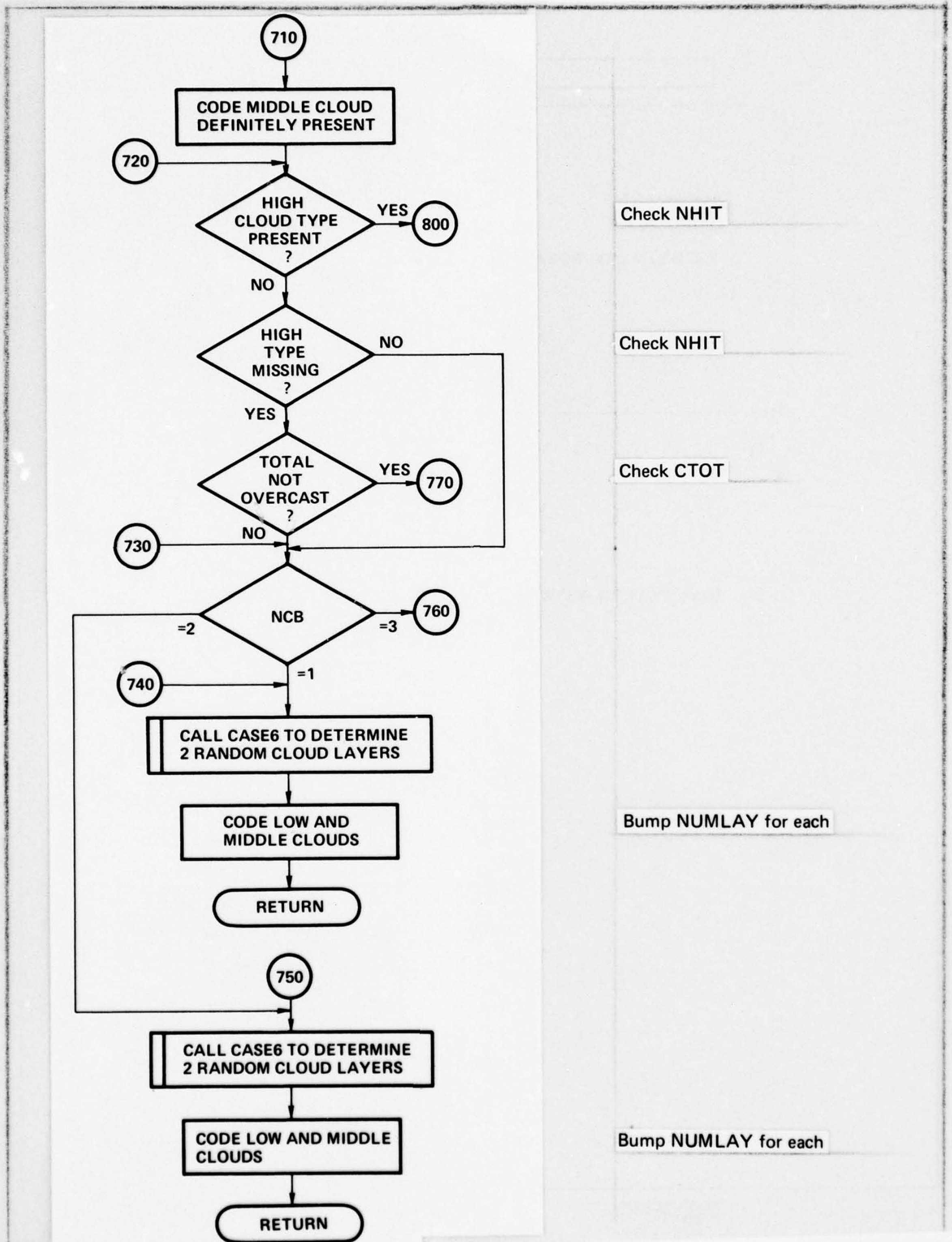
Bump NUMLAY

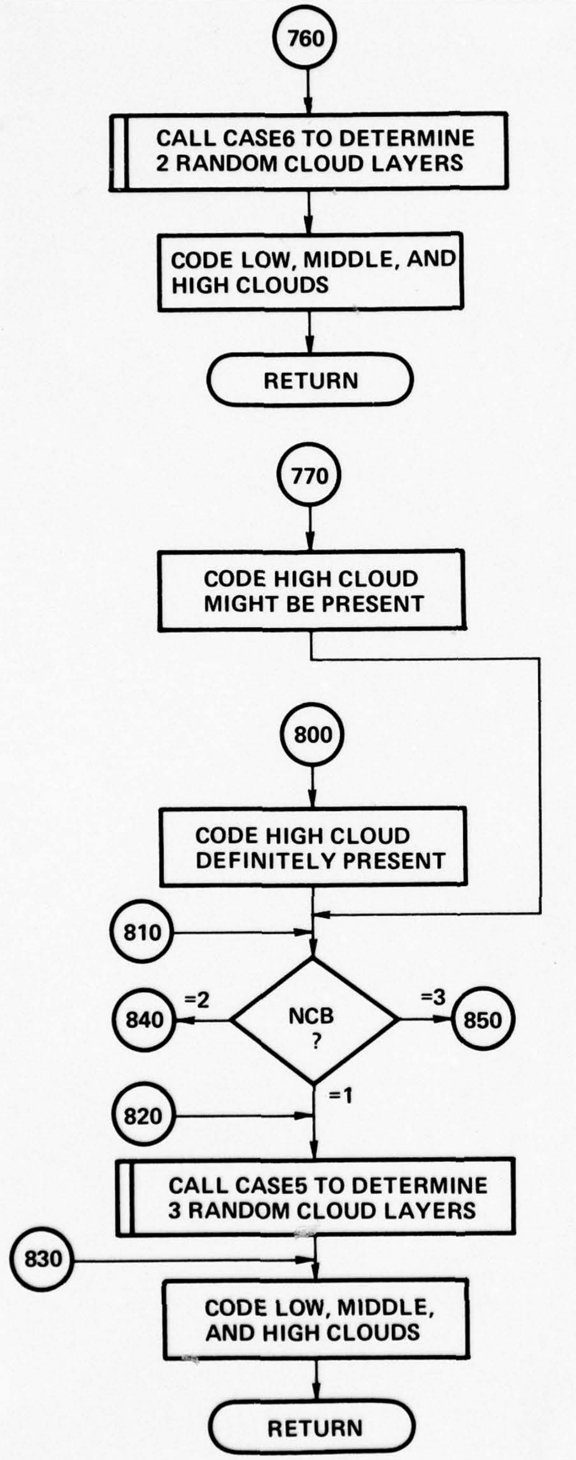
Bump NUMLAY

Check CTOT

Check NHIT

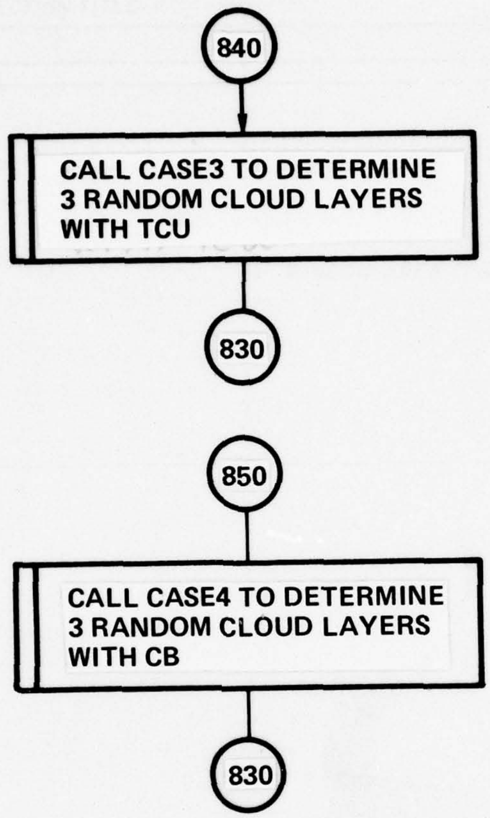

660
CODE LOW CLOUD
Bump NUMLAY
CALL CASE6 TO DETERMINE 2 RANDOM CLOUD LAYERS
CODE MIDDLE AND HIGH CLOUDS
Bump NUMLAY for each
TOTAL OVERCAST AND HIGH TYPE NOT GIVEN
YES
100
Check CTOT and NHIT
NO
670
RETURN
680
CODE LOW, MIDDLE AND HIGH CLOUDS
Bump NUMLAY for each
TOTAL OVERCAST AND HIGH TYPE NOT GIVEN ?
YES
100
Check CTOT and NHIT
NO
690
RETURN
700
CODE MIDDLE CLOUD TYPE MIGHT BE PRESENT
720


710
CODE MIDDLE CLOUD DEFINITELY PRESENT
720
HIGH CLOUD TYPE PRESENT ?
YES
800
NO
HIGH TYPE MISSING ?
NO
YES
TOTAL NOT OVERCAST ?
YES
770
NO
730
NCB
=2
=3
760
=1
740
CALL CASE6 TO DETERMINE 2 RANDOM CLOUD LAYERS
CODE LOW AND MIDDLE CLOUDS
RETURN
750
CALL CASE6 TO DETERMINE 2 RANDOM CLOUD LAYERS
CODE LOW AND MIDDLE CLOUDS
RETURN
Check NHIT
Check NHIT
Check CTOT
Bump NUMLAY for each
Bump NUMLAY for each



Bump NUMLAY for each

Bump NUMLAY for each



840
CALL CASE3 TO DETERMINE 3 RANDOM CLOUD LAYERS WITH TCU
830
850
CALL CASE4 TO DETERMINE 3 RANDOM CLOUD LAYERS WITH CB
830

SUBROUTINE TOPS (TERHT, NWEA, DLAT)

Routine to determine cloud tops given cloud bases, cloud cover, and weather.

TERHT	=	terrain height in feet
NWEA	=	weather in area (WMO code 4677)
WEAHIT	=	expected heights of cloud tops in 100's of feet due to weather
KCURW	=	weather factors for WX 50-99
KPWEA	=	weather factors WX 10-29
THICKO	=	thickness of cloud in feet at MSL
STHICK	=	slope of cloud thickness with respect to base of cloud above MSL
CLDTOP	=	maximum height of cloud top in feet
SAMT	=	conversion factor for cloud cover to cloud thickness factor
DLAT	=	latitude

Derived Layered Cloud Information

NUMLAY = number of layers generated

KIND = kind of cloud layer

- 1 = low
- 2 = middle
- 3 = high
- 4 = fog
- 5 = lowest cloud
- 6 = clear layer

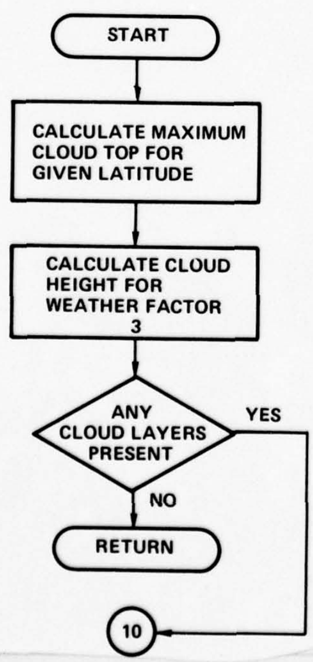
ITHIN = thin layer designator

- MISSING = not thin
- 1 = thin

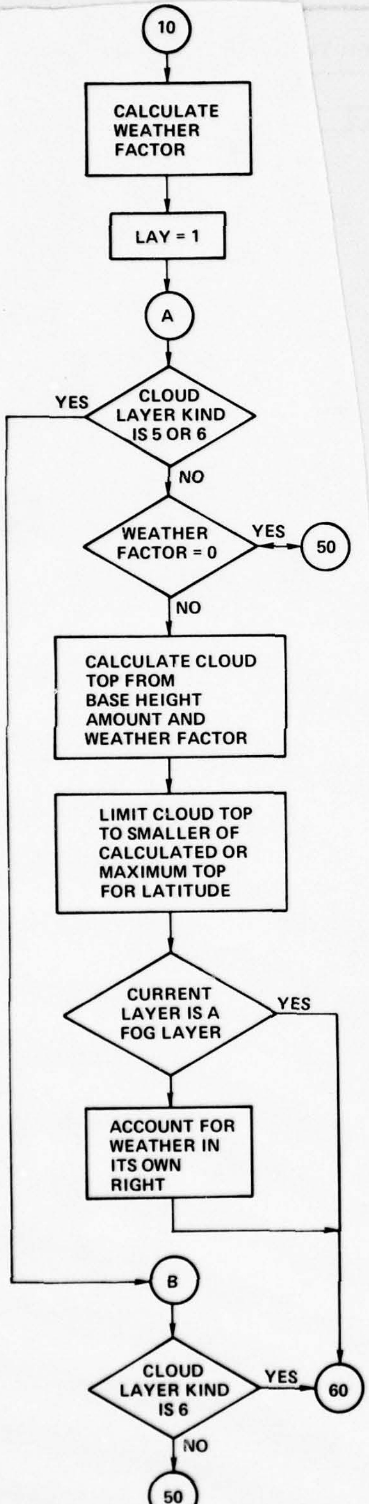
COVER = cloud cover in layer (0.0 - 1.0)

BASE = height of the base of layer, feet.

TOP = height of top of cloud layer, feet.



Maximum cloud height probable at latitude of OBS/REP



Initialize cloud layer index.

Jump to B if cloud layer KIND is not LOW, MIDDLE, HIGH or FOG.

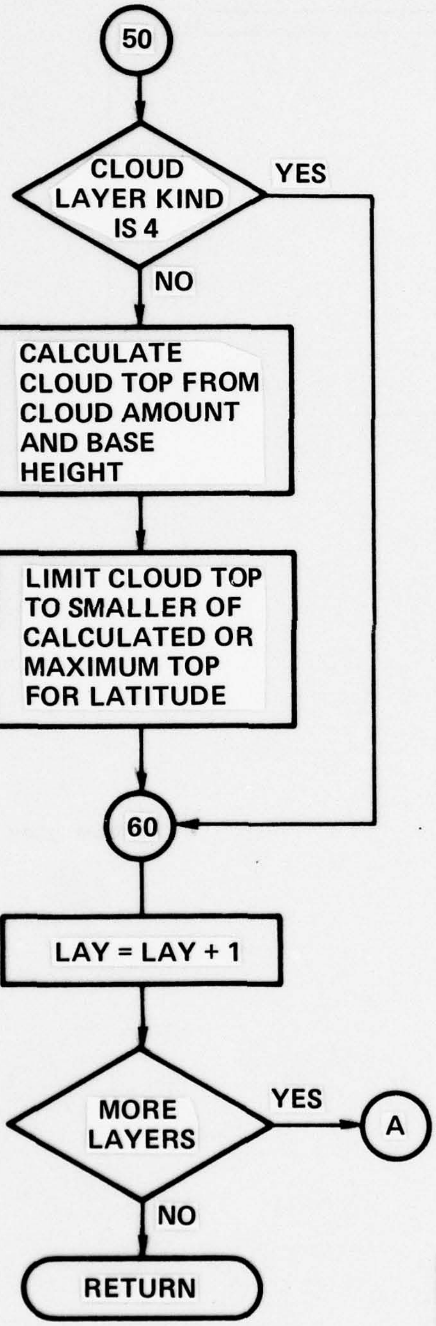
Calculation of cloud top from cloud amount, height of cloud layer base above mean sea level and non zero weather factor.

Cloud top cannot be greater than maximum probable height for latitude of OBS/REP

Jump to 60 if current layer is a FOG layer.

Set cloud top equal to larger of current value or value due to weather alone.

Jump to 60 if KIND of layer is CLEAR.



Jump to 60 if KIND of layer is FOG.

Calculation of cloud top from cloud amount and height of cloud layer base above mean sea level.

Cloud top cannot be greater than maximum probable height for latitude of OBS/REP.

SUBROUTINE UADINT

Routine to interpret upper air OBS/REP in terms of CFDB parameters

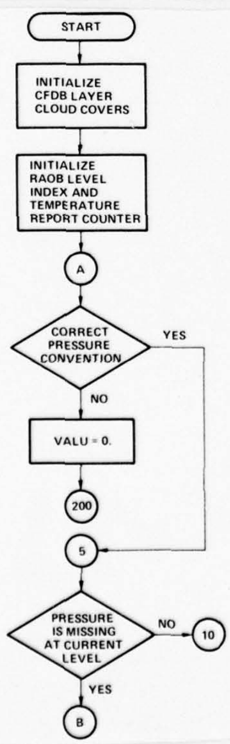
Sources of input data are upper air soundings (RAOBS) of pressure, temperature and dewpoint depression.

Input Data

IX = X distance of RAOB site from IXREF, hectometers.
IY = Y distance of RAOB site from IYREF, hectometers.
IH = Station elevation above mean sea level, meters.
ITIME = Time of RAOB, (0–1440)
ITYPE = 4, (–4 if a special RAOB)
IZ(I) = Altitude of RAOB reporting level, meters
IP(I) = Pressure of RAOB reporting levels, millibars*10
IT(I) = Temperature of RAOB reporting level, (deg. K.)*10
IDD(I) = Dewpoint depression of RAOB reporting level, (deg. C)*10
NRRL = Number of RAOB reporting levels

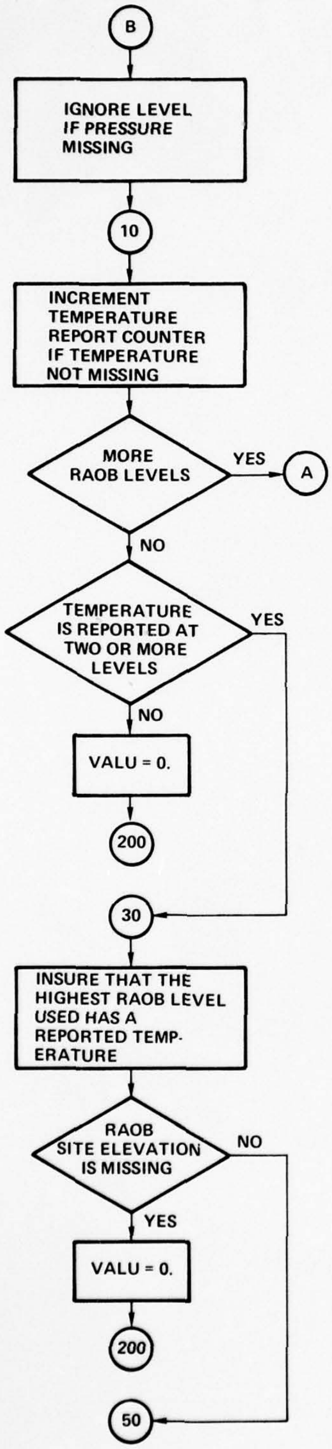
Cloud/fog data base parameters

IVALU = Information value of the RAOB (1–10)
0 = No CFDB parameters obtainable from the RAOB.
10 = No missing or inconsistent data in the RAOB.
0–10 = Some missing or inconsistent data in the RAOB.
MINBAS = Height of the base of the lowest cloud (AGL), dekameters.
MAXTOP = Height of the top of the highest cloud (AGL), dekameters.
LCOV(9) = Percent cloud cover in the CFDB layers.



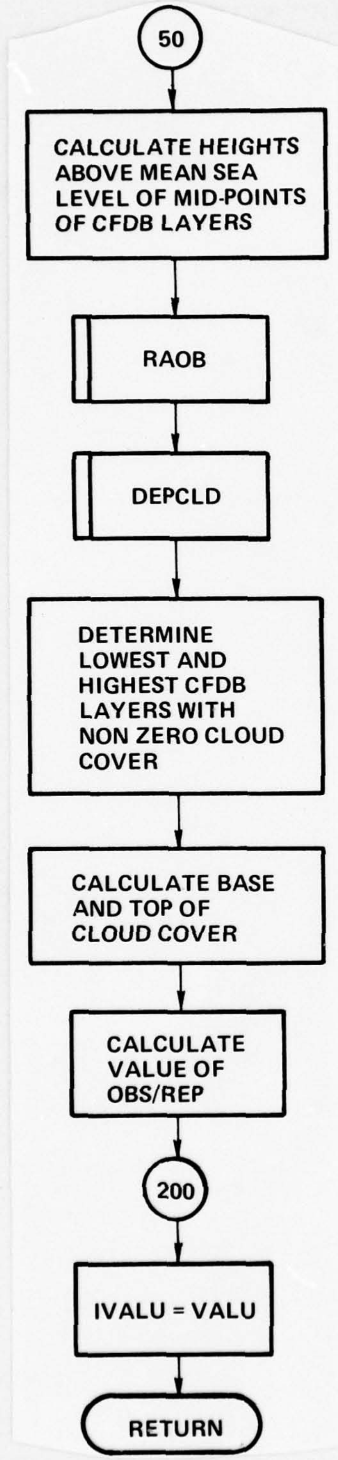
Pressure at the I+1 level must be less than the pressure at the I level.

If pressure convention incorrect, value is 0. Ignore report.



Ignore report and set value to 0 if temperature is not reported at two or more levels.

Ignore report and set value to 0 if RAOB site elevation is not given.



Determine temperature – dewpoint spreads at the midpoints of each of the CFDB layers.

Convert temperature – dewpoint spreads to cloud cover in each of the CFDB layers.

Base of cloud cover is base height of lowest CFDB layer with a non zero cloud cover. Top of cloud cover is the top height of the highest CFDB layer with a non zero cloud cover.

Use the fraction of the total number of CFDB layers for which cloud cover could not be determined and the value calculation in SUBROUTINE RAOB to determine OBS/REP value.

SECTION 4
OPERATING INSTRUCTIONS

4.1 TASKS AND DATA INPUTS

The CFAS is a subsystem to the EPAMS. The CFAS is called by the EPAMS through SUBROUTINE CFEXEC. Task requests and data are passed to the CFAS through the argument list in CFEXEC. The items in the argument list of CFEXEC are described in Table 2-6. The key element in this list is the integer variable, TASK, which tells the CFAS what task it is to perform. The particular task to be performed determines the other elements in the argument list for which values must be specified. A tabulation of these elements indicating those required for each of the four tasks is given in Table 4-1.

The OBS/REP are passed one at a time to CFAS on each call with TASK = 2 through the one dimensional array OBSRPT. The data elements in the five different types of OBS/REP are described in Tables 2-1 through 2-4. The ordering of these elements in each of the five types of OBS/REP is given in Table 4-2. Each element is of integer type.

The data which the user must supply through DATA and PARAMETER statements in SUBROUTINES CFEXEC, CFMAP and COMOBR are described in Table 2-6.

4.2 MASS STORAGE FILES

The CFAS requires access to five disk files having logical system file numbers 0 through 4. The size requirements of file number 0 is equal to the storage allocated to named COMMON/BASE/. The storage allocated to /BASE/ as well as the size requirements for logical system file numbers 1 and 2 depend upon the values of the storage/retrieval parameters set in SUBROUTINE BEGIN. A discussion of these parameters and their effect on running time and storage allocation is given in

TABLE 4-1

ELEMENTS IN THE ARGUMENT LIST OF CFEXEC REQUIRED IN EACH TASK

ELEMENT NO.	NAME	DIMENSION	TYPE	TASK 1	TASK 2	TASK 3	TASK 4
1	TASK	1	I	X	X	X	X
2	TIME	1	I			X	X
3	OBSRPT	143	I		X		
4	XO	1	FP				X
5	YO	1	FP				X
6	XLN	1	FP				X
7	YLN	1	FP				X
8	LAST	1	I			*	*
9	TYMOLD	1	I			X	X
10	DSP	1	FP			X	X
11	DIST	3	FP			X	X
12	TYMC	3	FP			X	X
13	ISSQ	5	I			X	X
14	NSSQ	1	I			X	X
15	NBKOUT	1	I			X	X
16	IDENT	10	I			X	X

* = Provide an address only for this variable in calling program.

I = integer.

FP = floating point.

-136-

TABLE 4-2

ORDERING OF ELEMENTS IN ARRAY OBSRPT

FOR THE FIVE OBS/REP TYPES

ELEMENT	TYPE ± 1	TYPE ± 2	TYPE ± 3	TYPE ± 4	TYPE 5
1	IX	IX	IX	IX	IX
2	IY	IY	IY	IY	IY
3	IZ	IZ	IZ	IH	IZ
4	ITIME	ITIME	ITIME	ITIME	ITIME
5	IOBC	IOBC	IOBC	IOBC	IOBC
6	ITYPE	ITYPE	ITYPE	ITYPE	ITYPE
7	IVALU	IVALU	IVALU	IVALU	IVALU
8					NTCLC
9					NCEIL
10					NVV
11					MINBAS
12					MAXTOP
13					MSPWE
14 - 22					LCOV(1-9)
23			ICL	IZ(1)	
24			ITSC	IZ(2)	
25			ICM	IZ(3)	
26			ICH	IZ(4)	
27 - 36	ICTS(1-10)	ICTS(1-10)	ICTS(1-10)	IZ(5-14)	
37 - 43	NWEA(1-7)	NWEA(1-7)	NWEA(1-7)	IZ(15-21)	
44			IPW	IZ(22)	
45	IDD	IDD	IDD	IZ(23)	
46	IFF	IFF	IFF	IZ(24)	
47	IPPP	IPPP	IPPP	IZ(25)	
48	ITT	ITT	ITT	IZ(26)	
49	ITD	ITD	ITD	IZ(27)	
50	IVIS	IVIS	IVIS	IZ(28)	

-137-

TABLE 4-2 (Continued)

ORDERING OF ELEMENTS IN ARRAY OBSRPT

FOR THE FIVE OBS/REP TYPES

ELEMENT	TYPE ± 1	TYPE ± 2	TYPE ± 3	TYPE ± 4	TYPE 5
51			NH	IZ(29)	
52			IH	IZ(30)	
53 - 62	NS(1-10)	NS(1-10)	NS(1-10)	IP(1-10)	
63 - 72	IHS(1-10)		IHS(1-10)	IP(11-20)	
73 - 82	ITHN(1-10)			IP(21-30)	
83	ICLG			IT(1)	
84	ICLGV			IT(2)	
85	IVISC			IT(3)	
86 - 112				IT(4-30)	
113 - 142				IDD(1-30)	
143				NRRL	

-138-

Section 3.3.1. The size of logical system file number 2 is equal to 23 times the number of OBS/REP used in a creation or update while the size of file number 4 is equal to the number of grid points in the CFDB times 15.

4.3 CORE REQUIREMENTS

Without segmentation and overlaying, CFAS requires approximately 12600_{10} words of instruction code and about 36300_{10} words for data for a 600 km. square window with a grid point spacing of 25 km. and with a dimension of 600 for the maximum number of OBS/REP to be used in a creation or update.

4.4 SAMPLE RUN

Using the test driver CFMAIN, Section 3.9, together with the system runstream elements .TR001C and .STORE and data elements .TR001D and .OBSREP (all listed in Appendix I), the created CFDB shown in Fig. 4-1 and the update CFDB shown in Fig. 4-2 are produced.

```
CFAS SAMPLE RUN                                                                    DATE 111375      PAGE    3

     TASK=  3     NPRT=   0     TIME=  160

   TIME      TYMOLD      NSSQ     ISSQ(1)   ISSQ(2)   ISSQ(3)   ISSQ(4)
    160       1475          4         1         2         3         4

    DSP       DIST(1)      DIST(2)      DIST(3)      TYMC(1)      TYMC(2)      TYMC(3)
    20.0         20.0         80.0        100.0         50.0        120.0        150.0
```

FIG. 4-1 Created CFDB.

CFAS SAMPLE RUN DATE 111375 PAGE 4

CONTENTS OF BLOCK NO. 1 OF THE CFDB FILE

IDENT= 0 0 0 0 0 0 0 0 0 0

GRID POINT DATA FOLLOWS

I	J	SKYC	CEIL	VIS	BASE	TOP	WTHR	LAY1	LAY2	LAY3	LAY4	LAY5	LAY6	LAY7	LAY8	LAY9
1	1	91	-32768	900	0	1105	85	25	25	25	25	25	25	25	0	60
1	2	93	701	2760	29	1105	85	15	15	15	15	15	20	20	5	40
1	3	99	701	1938	17	303	43	75	75	0	0	0	5	5	10	5
1	4	100	701	801	0	5	43	100	100	0	0	0	0	0	5	0
1	5	100	701	2000	67	165	35	0	0	0	0	0	100	100	100	0
1	6	63	701	2100	175	1103	90	0	0	0	0	0	0	0	50	50
1	7	65	701	1833	136	1103	90	5	5	5	5	5	5	5	50	50
1	8	72	701	900	0	1101	90	25	25	25	25	25	25	25	50	50
2	1	95	-32768	2059	32	1105	63	10	10	10	10	10	20	20	10	35
2	2	98	701	3007	47	1050	63	5	5	5	5	5	20	15	20	30
2	3	99	701	2058	33	598	63	45	45	0	0	0	10	10	15	15
2	4	100	701	801	0	5	43	100	100	0	0	0	0	0	5	0
2	5	65	701	1611	140	927	90	0	0	0	0	0	0	65	65	65
2	6	61	701	1744	165	1019	90	0	0	0	0	0	0	70	50	55
2	7	66	701	1403	129	915	90	20	20	20	20	20	20	30	50	55
2	8	65	701	918	0	880	90	35	30	30	30	30	30	30	50	45
3	1	100	-32768	5000	64	1105	63	0	0	0	0	0	15	15	15	15
3	2	80	501	3221	73	877	63	5	5	5	5	5	15	5	35	35
3	3	75	701	2424	77	890	63	5	5	5	5	5	20	15	40	40
3	4	75	701	800	80	1103	90	0	0	0	0	0	50	50	50	50
3	5	73	701	1397	124	975	90	0	0	0	0	0	15	60	50	60
3	6	70	701	1549	152	955	90	0	0	0	0	0	0	55	55	60
3	7	85	701	1014	57	974	95	25	25	25	25	25	25	40	40	55
3	8	95	701	929	0	911	95	35	30	30	30	30	30	30	30	45
4	1	77	-32768	2520	55	945	63	20	20	20	20	25	30	25	45	35
4	2	57	442	1093	87	706	55	10	10	10	10	15	15	5	40	40
4	3	53	476	1443	36	899	90	5	5	5	5	10	25	20	45	40
4	4	84	701	1044	51	814	90	0	0	0	35	35	65	65	65	65
4	5	93	701	1095	60	793	90	20	20	20	55	45	55	70	70	70
4	6	93	701	1059	85	998	82	40	40	40	40	20	20	60	60	60
4	7	93	701	827	7	814	95	40	40	40	40	40	40	45	45	60
4	8	99	701	827	7	924	95	40	40	40	40	40	40	40	40	60
5	1	84	701	900	0	1105	90	75	75	75	75	75	75	75	75	35
5	2	74	450	1262	18	1105	90	50	50	50	50	60	60	60	60	30
5	3	63	400	1247	23	1104	80	25	25	25	25	35	35	35	35	35
5	4	90	374	1311	26	733	50	15	15	15	60	65	65	70	70	70
5	5	100	373	1325	23	732	50	25	25	25	70	65	65	70	70	70
5	6	70	382	1705	30	1102	82	45	45	45	45	50	50	50	50	50
5	7	100	333	1332	17	782	95	45	45	45	45	60	60	50	50	70
5	8	90	701	623	26	736	75	40	40	40	40	65	65	70	70	75
6	1	73	701	1294	14	1105	90	60	60	60	60	50	70	70	70	40
6	2	70	701	1209	12	1105	90	60	60	60	60	60	70	70	70	40
6	3	75	394	400	0	1104	80	50	50	50	50	50	50	50	50	50

FIG. 4-1 Created CFDB (cont.).

CFAS SAMPLE RUN DATE 111375 PAGE 5

I	J	SKYC	CEIL	VIS	BASE	TOP	WTHR	LAY1	LAY2	LAY3	LAY4	LAY5	LAY6	LAY7	LAY8	LAY9
6	4	75	182	722	11	1104	80	40	40	40	40	45	45	50	50	50
6	5	100	182	1744	49	773	61	0	0	0	0	35	35	45	85	85
6	6	45	182	1866	57	973	61	0	0	0	0	20	30	30	45	45
6	7	22	182	1367	57	1101	63	5	5	5	5	10	20	20	20	20
6	8	32	316	763	43	807	75	10	10	10	10	55	60	50	50	60
7	1	60	701	2543	57	1105	91	20	20	20	20	20	55	55	55	45
7	2	62	701	2207	43	1105	91	25	25	25	25	25	55	55	55	45
7	3	75	701	400	0	1104	80	50	50	50	50	50	50	50	50	50
7	4	78	182	380	0	1009	80	55	55	45	45	45	55	55	55	45
7	5	100	182	1541	44	519	61	5	5	0	0	40	45	45	100	95
7	6	45	182	1567	57	397	61	0	0	0	0	15	25	25	45	45
7	7	22	182	1685	57	1004	63	10	10	10	10	5	15	15	15	15
7	8	77	182	1073	69	918	75	5	5	5	5	35	40	55	55	55
8	1	50	701	2244	133	1105	**	0	0	0	0	0	20	20	45	45
8	2	50	701	2252	132	1105	**	0	0	0	0	0	20	20	45	45
8	3	75	701	400	0	1104	80	50	50	50	50	50	50	50	50	50
8	4	79	182	385	0	959	80	55	55	40	40	40	60	50	50	40
8	5	58	182	299	0	1064	63	25	25	25	25	25	20	20	20	15
8	6	56	182	239	0	1102	63	25	25	25	25	25	15	15	15	15
8	7	57	182	319	0	1071	63	25	25	25	25	25	20	20	20	15
8	8	100	182	1574	133	893	90	0	0	0	0	0	5	55	55	60

FIG. 4-1 Created CFDB (cont.).

```
CFAS SAMPLE RUN                                                                          DATE 111375        PAGE     6

    TASK=  4      NPRT=   0      TIME=  170

  TIME      TYMOLD      NSSG     ISSG(1)    ISSG(2)    ISSG(3)    ISSG(4)
   170       1425          4           1          2          3          4

    DSP       DIST(1)      DIST(2)      DIST(3)      TYMC(1)      TYMC(2)      TYMC(3)
    20.0         20.0         80.0        100.0         50.0        120.0        150.0

    XO          YO          XLN          YLN
    51.0        51.0        99.0         99.0
```

FIG. 4-2 Update CFDB.

CFAS SAMPLE RUN   DATE 111375   PAGE 7

CONTENTS OF BLOCK NO. 2 OF THE CFDB FILE

IDENT= 0 0 0 0 0 0 0 0 0 0

GRID POINT DATA FOLLOWS

I	J	SKYC	CEIL	VIS	BASE	TOP	WTHR	LAY1	LAY2	LAY3	LAY4	LAY5	LAY6	LAY7	LAY8	LAY9
3	3	75	701	2423	76	893	63	5	5	5	5	5	20	15	40	40
3	4	75	701	800	80	1103	90	0	0	0	0	0	50	50	50	50
3	5	77	701	1402	125	974	90	0	0	0	0	0	15	50	50	60
3	6	69	701	1550	152	955	90	0	0	0	0	0	0	55	55	60
3	7	85	701	1008	55	943	95	25	25	25	25	25	25	40	40	55
4	3	59	470	1429	85	889	90	5	5	5	5	10	25	20	45	40
4	4	83	701	1033	62	827	90	0	0	0	35	35	65	55	65	65
4	5	92	701	1089	59	793	90	20	20	20	55	45	55	70	70	70
4	6	93	701	1043	34	999	82	40	40	40	40	20	20	50	60	60
4	7	98	701	823	7	916	95	40	40	40	40	40	40	45	45	60
5	3	62	402	1250	23	1174	55	25	25	25	25	35	35	35	35	35
5	4	90	326	1301	25	744	50	15	15	15	60	65	65	70	70	70
5	5	100	356	1333	28	736	50	25	25	25	70	65	65	70	70	70
5	6	69	365	1709	30	1102	82	45	45	45	45	50	50	50	50	50
5	7	100	365	1315	15	776	95	50	50	50	50	65	55	55	65	70
6	3	75	389	400	0	1104	80	50	50	50	50	50	50	50	50	50
6	4	75	182	713	11	1174	80	40	40	40	40	45	45	50	50	50
6	5	100	182	1739	49	726	61	0	0	0	0	35	35	45	85	85
6	6	50	182	1353	57	918	61	0	0	0	0	20	30	30	50	50
6	7	22	182	1877	57	1101	63	5	5	5	5	10	20	20	20	20
7	3	75	701	400	0	1174	30	50	50	50	50	50	50	50	50	50
7	4	77	182	393	0	1040	80	50	50	45	45	45	55	55	55	45
7	5	100	182	1555	44	526	61	5	5	0	0	40	40	40	100	95
7	6	45	182	1650	57	893	61	0	0	0	0	15	25	25	45	45
7	7	22	182	1687	57	1007	63	10	10	10	10	5	15	15	15	15

EXECUTION TERMINATED BY AN ATTEMPT TO READ PAST AN END-OF-FILE.

@FIN

FIG. 4-2 Update CFDB (cont.).

SECTION 5
CONCLUSIONS

The design concept of the CFAS that has evolved in this effort is reasonably close to that which was envisioned at the outset. The CFAS can perform the intended function of creating and maintaining a cloud, fog and weather data base in near real time. The actual total execution time on a Univac 1106 computer is approximately 38 seconds for a creation made on a 600 km. square window of grid points spaced 25 km. apart from seventy OBS/REP. The core storage requirements turned out to be somwhat larger than originally planned, but the CFAS is structured for convenient segmentation and overlaying which can substantially reduce the core storage requirements at a modest cost in execution time.

The design of the CFAS is such that provisions for the evaluation and interpretation of cloud and fog information from data sources other than those currently employed can be easily incorporated. This feature was deemed to be particularly important because it became evident early in the program that the density of conventional surface and upper air observations could be rather sparse in certain operational environments.

Our recommendations for further efforts on the CFAS are contained in Section 6.

SECTION 6
RECOMMENDATIONS

Our principal recommendation is that the CFAS be subjected to a test and evaluation program using an historical data base representative of the various meteorological regimes in which the CFAS must function. Verification schemes should be employed in the evaluation which are free of any bias and designed to evaluate the analysis error for the entire analysis area as well as at discrete points. Most important in the evaluation should be a determination of the affect of data density and its variability.

Other recommendations include the following:

1) An investigation of the cost/benefit of additional data sources such as radar, satellite, and the more or less qualitative but numerous observations that could be obtained from non-meteorological personnel deployed in the field Army's region of responsibility.
2) An experimental determination of suitable values for the OBS/REP storage and retrieval parameters. This should be a coordinated effort with other EPAMS activities and done as part of an optimization of the CFAS-EPAMS interaction.
3) An investigation of improved probabilistical and statistical techniques for the inference of cloud parameters.

REFERENCES

1. A. R. Coburn, Major USAF AFGWC, "Improved Three Dimensional Nephanalysis Model," Tech. Mem., pp. 71-2, June 1971.

2. W. D. Mount, B. Kunkel, and S. J. Penn, "An Objective Analysis of Continuous and Discontinuous Parameters," Appl. Meteorology 2, p. 345, (1963).

3. S. L. Barnes, "Mesoscale Objective Map Analysis Using Weighted Time-Series Observations," NOAA TM ERL NSSL-62, March 1973.

4. E. L. Davis, "Objective Techniques for the Analysis of Clouds and Ceilings," FAA Contract FAA/BRD-363, Technical Publication 18, November 1962.

5. H. Edson, "Numerical Cloud and Icing Forecasts," Services Technical Note 13, H.Q. 3rd Weather Wing, September 1965.

APPENDIX I

PROGRAM CODE LISTINGS

CFAS SUBPROGRAM ELEMENT AFDINT

CLOUD-FOG*CFAS.AFDINT

```
          SUBROUTINE AFDINT
          COMMON /OBSREP/ IX,IY,IZ,ITIME,IOBC,ITYPE,IVALU,NTCLC,NCEIL,NVV
         *,MINBAS,MAXTOP,MSPWE,LCOV(15),NOUSE(115)
          DIMENSION LCOVB(16)
          DO 10 I=1,9
       10 LCOVB(I)=LCOV(I)
          Z=IZ*3.281
          IF(Z .LE. 150.) GO TO 100
          IF(Z .GT. 1500.) GO TO 30
          DO 20 I=1,3
       20 LCOVB(I+6)=MAXO(LCOV(I+6),LCOV(I+7))
          GO TO 100
       30 IF(Z .GT. 1650.) GO TO 40
          LCOVB(7)=LCOV(8)
          LCOVB(9)=MAXO(LCOV(9),LCOV(10))
          GO TO 100
       40 IF(Z .GT. 3300.) GO TO 50
          LCOVB(7)=LCOV(9)
          LCOVB(8)=LCOV(9)
          LCOVB(9)=LCOV(10)
          GO TO 100
       50 IF(Z .GT. 6500.) GO TO 60
          LCOVB(7)=MAXO(LCOV(9),LCOV(10))
          LCOVB(8)=LCOVB(7)
          LCOVB(9)=LCOV(10)
          GO TO 100
       60 LCOVB(7)=LCOV(10)
          LCOVB(8)=LCOVB(7)
          LCOVB(9)=MAXO(LCOV(10),LCOV(11))
      100 DO 110 I=1,9
      110 LCOV(I)=LCOVB(I)
          RETURN
          END
```

@HDG,P CFAS SUBPROGRAM ELEMENT BAKUTM

@PRT,S CFAS.BAKUTM
FURPUR 0026-10/28-13:57

CFAS SUBPROGRAM ELEMENT BAKUTM

```
CLOUD-FOG*CFAS.BAKUTM
            SUBROUTINE BAKUTM (W,Z,X,Y,CMRD)
      C
      C  INVERSE OF UTM - CONVERTS HUNDREDS OF KILOMETERS TO DEGREES.
      C       A - CONVERSION FACTOR (100'S OF KM/RADIAN ALONG GREAT CIRCLE)
      C     RAD - CONVERSION FACTOR (RADIAN/DEGREE)
      C    CMRD - CENTRAL MERIADIAN IN DEGREES
      C    DWN, DZN, W, WN, Z, ZN - IN DEGREES
      C      DX, DY, X, XN, Y, YN - IN 100'S OF KM
      C
            A = 63.782064
            RAD = 0.017453292
            ZN = (Y/(A * RAD))
            WN = (-X/(A * COS(ZN * RAD) * RAD)) + CMRD
            DO 10 I = 1,10
            CALL UTM (WN,ZN,XN,YN,CMRD)
            DX = X - XN
            DY = Y - YN
            DZN = (DY/(A * RAD))
            ZN = ZN + DZN
            DWN = (-DX/(A * COS((ZN + DZN) * RAD) * RAD))
            WN = WN + DWN
            IF (ABS(DX) .LT. 1.E-5 .AND. ABS(DY) .LT. 1.E-5) GO TO 20
      10    CONTINUE
      20    CONTINUE
            W = WN
            Z = ZN
            RETURN
            END
```

@HDG,P CFAS SUBPROGRAM ELEMENT BEGIN

@PRT,S CFAS.BEGIN
FURPUR 0026-10/28-13:57

CFAS SUBPROGRAM ELEMENT BEGIN

CLOUD-FOG*CFAS.BEGIN

```
      SUBROUTINE BEGIN
C NOTE - UNLESS OTHERWISE NOTED - ALL DISTANCE MEASUREMENTS, UTM UNITS,
C AND UTM COORDINATES ARE CARRIED IN HECTOMETERS WHERE 1 HECTOMETER
C EQUALS 100 METERS.
C NOTE - UNLESS OTHERWISE NOTED - ALL TIMES WILL BE CARRIED IN MINUTES
C FOR A 1440 MINUTE CLOCK.
C
C XREF AND YREF MUST BE IN KILOMETERS AND MUST BE SUPPLIED BY THE
C CALLING PROGRAM.
      COMMON /MAP/ XREF, YREF, CMRD
C
      COMMON /BASE/ DXSECT, DYSECT, EDGE, IBLOCK, IDTIME, IDXUTM,
     * IDYUTM, INUMBR, ISTATI, ISTATO, JNUMBR, JSTATI, JSTATO, JTIME,
     * LASTJ, MAXGPS, NBJNOW, NBLKFJ, NCOLS, NGX, NGY, NINI, NINTAB,
     * NROWS, NRPBFI, NRPBFJ, NSECTR, NWDBKI, NWDBKJ, NWDREC, NXSECT,
     * NYSECT, UTMPGD, XBASE, XMAX, XMIN, YBASE, YMAX, YMIN,
     * NNEWRS(100), NALLRS(100), ITABLE(4, 500), IBUF(3750), JBUF(1000),
     * JTIMES(100), IRXMAX, IRXMIN, IRYMAX, IRYMIN
C            NROWS = NO. OF ROWS IN GRID.
      NROWS=24
C            NCOLS = NO. OF COLUMNS IN GRID.
      NCOLS=24
C            UTMPGD = HECTOMETERS PER GRID UNIT.
      UTMPGD=250.0
C            XBASE = ABSOLUTE EAST-WEST UTM GRID COORDINATE OF LOWER
C            LEFT CORNER OF GRID IN HECTOMETERS.
      XBASE=XREF*10.0
C            YBASE = ABSOLUTE NORTH-SOUTH UTM GRID COORDINATE OF LOWER
C            LEFT CORNER OF GRID IN HECTOMETERS.
      YBASE=YREF*10.0
C            EDGE = MINIMUM DISTANCE FROM OUTSIDE GRID POINTS TO
C            OUTER BOARDER OF OUTSIDE STORAGE SECTOR IN HECTOMETERS.
      EDGE=500.0
C            MAXGPS = MAXIMUM NO. OF GRID POINTS PER STORAGE SECTOR.
      MAXGPS=64
C            NWDREC = NO. OF WORDS PER OBS/REP RECORD.
      NWDREC=44
C            NRPBFI = NO. OF RECORDS PER BLOCK IN FILE I.
      NRPBFI=85
C            NRPBFJ = NO. OF RECORDS PER BLOCK IN FILE J.
      NRPBFJ=22
C            NBLKFJ = NO. OF BLOCKS IN FILE J.
      NBLKFJ=25
C            NINTAB = NO. OF COLUMNS IN ITABLE.
      NINTAB=500
C            NWDBKJ = NO. OF WORDS PER BLOCK IN FILE J.
      NWDBKJ=NWDREC*NRPBFJ
C            NWDBKI = NO. OF WORDS PER BLOCK IN FILE I.
      NWDBKI=NWDREC*NRPBFI
C            NBJNOW = NO. OF BLOCKS IN FILE J WHICH NOW CONTAIN OLD
C            DATA RECORDS.
      NBJNOW=0
C            NINI = NO. OF ENTRIES IN ITABLE NOW.
      NINI=0
C            IBLOCK = BLOCK NO. OF BLOCK IN FILE I THAT IS NOW IN CORE.
      IBLOCK=0
C            INUMBR = FILE NO. OF FILE I.
      INUMBR=1
```

```
CFAS SUBPROGRAM ELEMENT BEGIN

      C              JNUMBR = FILE NO. OF FILE J.
             JNUMBR=2
      C              IDTIME = WORD IN DATA RECORD CONTAINING TIME OF OBS/REP IN
      C              MINUTES (0 - 1439).
             IDTIME=4
      C              IDXUTM = WORD IN DATA RECORD CONTAINING RELATIVE X POSITION
      C              OF OBS/REP.
             IDXUTM=1
      C              IDYUTM = WORD IN DATA RECORD CONTAINING RELATIVE Y POSITION
      C              OF OBS/REP.
             IDYUTM=2
             XBKILO=XBASE/10.0
             YBKILO=YBASE/10.0
             PRINT 500 XBKILO, YBKILO
         500 FORMAT (1H , ' BEGIN - UTM COORDINATES OF LOWER LEFT HAND CORNER O
            *F WINDOW IN KILOMETERS ARE X =', F9.2, '    Y =', F9.2)
             PRINT 510 NROWS, NCOLS, UTMPCD
         510 FORMAT (1H , ' BEGIN - GRID CONTAINS', I4, ' ROWS AND', I4, ' COLU
            *MNS WITH A GRID INTERVAL OF', F8.2, ' HECTOMETERS')
             PRINT 520 NWDREC, IDXUTM, IDYUTM, IDTIME
         520 FORMAT (1H , ' BEGIN - OBS/REP RECORDS WILL CONTAIN', I4, ' WORDS
            *WITH ---', /,
            * 10X, 'WORD NO.', I2, ' = RELATIVE X COORDINATE IN HECTOMETERS',/,
            * 10X, 'WORD NO.', I2, ' = RELATIVE Y COORDINATE IN HECTOMETERS',/,
            * 10X, 'WORD NO.', I2, ' = TIME IN MINUTES (0 - 1439)')
             CALL SECTOR
             DO 10 I=1, 100
             NNEWRS(I)=0
          10 NALLRS(I)=0
             RETURN
             END

BHDG,P CFAS SUBPROGRAM ELEMENT BLKIN

@PRT,S CFAS.BLKIN
FURPUR 0026-10/28-13:57
```

BEST AVAILABLE COPY

CFAS SUBPROGRAM ELEMENT BLKIN

```
CLOUD-FOG*CFAS.BLKIN
      SUBROUTINE BLKIN (NWDBLK, ISTART, NBKIN, LSFILE, ISTAT)

C     DISK VERSION. BLOCK TRANSFER FROM RANDOM ACCESS DISK FILE TO CORE.

C BLKIN TRANSFERS TO CORE A BLOCK FROM A RANDOM ACCESS FILE THAT
C CONTAINS BLOCKS THAT ARE ALL OF THE SAME SIZE.
C NWDBLK = NO. OF WORDS PER BLOCK IN THE FILE AND THE NO. OF WORDS TO BE
C          TRANSFERRED TO CORE ON THIS CALL.
C ISTART = STARTING ADDRESS IN CORE WHERE THE BLOCK IS TO BE TRANSFERRED
C          TO.
C NBKIN  = NO. OF THIS BLOCK IN THE FILE.  NBKIN = 1 IS THE FIRST BLOCK
C          NO. IN THE FILE.
C LSFILE = LOGICAL SYSTEM FILE NO. (0-15).
C ISTAT  = STATUS RETURNED TO USER.  ISTAT = 0 INDICATES NO ERRORS.
C          ISTAT = 1 INDICATES AN ERROR OF SOME KIND.
C
C 1108 DISK VERSION
C
C RESTRICTIONS ON THIS VERSION OF BLKIN
C---
C THE STATUS ISTAT RETURNED TO THE USER WILL ALWAYS BE ZERO SINCE THE
C FSTRD ROUTINE DOES NOT RETURN ANY STATUS INFORMATION.  FSTRD HAS IT'S
C OWN ERROR MESSAGES.
C
      NSECPB=(NWDBLK+27)/28
      NBKM1=NBKIN-1
      CALL FSTRD (NWDBLK, ISTART, NSECPB, NBKM1, 0, LSFILE)
      ISTAT=0
      RETURN
      END
```

@HDG,P CFAS SUBPROGRAM ELEMENT BLKOUT

@PRT,S CFAS.BLKOUT
FURPUR 0026-10/28-13:57

CFAS SUBPROGRAM ELEMENT BLKOUT

```
CLOUD-FOG*CFAS.BLKOUT
            SUBROUTINE BLKOUT (NWDBLK, ISTART, NBKOUT, LSFILE, ISTAT)

      C     DISK VERSION. BLOCK TRANSFER FROM CORE TO RANDOM ACCESS DISK FILE.

      C BLKOUT TRANSFERS A BLOCK FROM CORE TO A RANDOM ACCESS FILE WHICH
      C CONTAINS BLOCKS THAT ARE ALL OF THE SAME SIZE.
      C NWDBLK = NO. OF WORDS PER BLOCK IN THE FILE AND THE NO. OF WORDS TO BE
      C          TRANSFERRED FROM CORE ON THIS CALL.
      C ISTART = STARTING ADDRESS IN CORE WHERE THE BLOCK IS TO BE TRANSFERRED
      C          FROM.
      C NBKOUT = NO. OF THIS BLOCK IN THE FILE.  NBKOUT = 1 IS THE FIRST BLOCK
      C          NO. IN THE FILE.
      C LSFILE = LOGICAL SYSTEM FILE NO. (0-15).
      C ISTAT  = STATUS RETURNED TO USER.  ISTAT = 0 INDICATES NO ERRORS.
      C          ISTAT = 1 INDICATES AN ERROR OF SOME KIND.
      C
      C 1108 DISK VERSION
      C
      C RESTRICTIONS ON THIS VERSION OF BLKOUT
      C
      C THE STATUS ISTAT RETURNED TO THE USER WILL ALWAYS BE ZERO SINCE THE
      C FSTWT ROUTINE DOES NOT RETURN ANY STATUS INFORMATION.  FSTWT HAS IT'S
      C OWN ERROR MESSAGES.
      C
            NSECPB=(NWDBLK+27)/28
            NBKM1=NBKOUT-1
            CALL FSTWT (NWDBLK, ISTART, NSECPB, NBKM1, 0, LSFILE)
            ISTAT=0
            RETURN
            END

@HDG,P CFAS SUBPROGRAM ELEMENT CASES

@PRT,S CFAS.CASES
FURPUR 0026-10/28-13:57
```

CFAS SUBPROGRAM ELEMENT CASES

CLOUD-FOG*CFAS.CASES

```
      SUBROUTINE CASE1(G1,G2,G3,CTOT,CLD1,CLD2,CLD3)
C
C     ROUTINE TO CALCULATE THREE LAYERS OF CLOUD COVER GIVEN TOTAL
C           CLOUD COVER ASSUMING LAYERS ARE COMPLETELY RANDOM.
C     G1 = PROBABILITY OF RANDOM CLOUD IN LAYER 1
C     G2 = PROBABILITY OF RANDOM CLOUD IN LAYER 2
C     G3 = PROBABILITY OF RANDOM CLOUD IN LAYER 3
C     CTOT = TOTAL CLOUD COVER   (RANGE 0 - 1)
C     CDL1 = CLOUD COVER OF LAYER 1 (RANGE 0 - 1)
C     CDL2 = CLOUD COVER OF LAYER 2 (RANGE 0 - 1)
C     CDL3 = CLOUD COVER OF LAYER 3 (RANGE 0 - 1)
C
C     INITIALIZE INTERMEDIATE CLOUD COVER AND INTERMEDIATE FACTORS.
C
      CLD=CTOT
      GP=G1*G2*G3
      GS=G1+G2+G3
      GSP=G1*G2+G1*G3+G2*G3
C
C     USE ITERATIVE SOLUTION.
C
    1 FUNCLD=GP*CLD**3-GSP*CLD**2+GS*CLD-CTOT
      DELFUN=3.*GP*CLD**2-2.*GSP*CLD+GS
      IF(ABS(DELFUN).GT.0.0001) GO TO 2
      DELFUN=SIGN(0.0001,DELFUN)
    2 DELCLD=FUNCLD/DELFUN
      CLD=CLD-DELCLD
C
C     REITERATE IF CHANGE IN INTERMEDIATE CLOUD COVER UNACCEPTABLE.
C
      IF(ABS(DELCLD).GT.0.01) GO TO 1
C
C     CALCULATE LAYERED CLOUD COVER.
C
      CLD1=G1*CLD
      CLD2=G2*CLD
      CLD3=G3*CLD
      RETURN
C
      ENTRY CASE2(G1,G2,CTOT,CLD1,CLD2)
C
C     ROUTINE TO CALCULATE TWO LAYERS OF CLOUD COVER GIVEN TOTAL
C          CLOUD COVER ASSUMING LAYERS ARE COMPLETELY RANDOM.

C     G1 = PROBABILITY OF RANDOM CLOUD IN LAYER 1
C     G2 = PROBABILITY OF RANDOM CLOUD IN LAYER 2
C     CTOT = TOTAL CLOUD COVER (RANGE 0 - 1)
C     CLD1 = CLOUD COVER OF LAYER 1 (RANGE 0 - 1)
C     CLD2 = CLOUD COVER OF LAYER 2 (RANGE 0 - 1)

C     CALCULATE INTERMEDIATE FACTORS.

      GP=G1*G2
      GS=G1+G2
      CLD=(GS-SQRT(GS**2-4.*GP*CTOT))/(2.*GP)

C     CALCULATE LAYERED CLOUD COVER.

```

```
              CLD1=G1*CLD
              CLD2=G2*CLD
              RETURN

              ENTRY CASE3(G2,G3,CLOW,CTOT,CLD1,CLD2,CLD3,REDUCE)

      C       ROUTINE TO CLACULATE THREE LAYERS OF CLOUD COVER GIVEN LOWEST
      C            CLOUD COVER AND TOTAL CLOUD COVER ASSUMING A TCU IN LAYERS
      C             AND 2 WITH RANDOM LAYERS 2 AND 3

      C       G2 = PROBABILITY OF RANDOM CLOUD IN LAYER 2
      C       G3 = PROBABILITY OF RANDOM CLOUD IN LAYER 3
      C       CLOW = CLOUD COVER OF TCU (RANGE 0 - 1)
      C       CTOT = TOTAL CLOUD COVER (RANGE 0 - 1)
      C       CLD1 = CLOUD COVER OF LAYER 1 (RANGE 0 - 1)
      C       CLD2 = CLOUD COVER OF LAYER 2 (RANGE 0 - 1)
      C       CLD3 = CLOUD COVER OF LAYER 3 (RANGE 0 - 1)
      C       REDUCE = TCU REDUCTION FACTOR
      C
      C       CALCULATE INTERMEDIATE FACTORS.
      C
              GS=(G2+G3)*(1.-CLOW)
              GP=G2*G3*(1.-CLOW)
              CLD=(GS-SQRT(GS**2-4*GP*(CTOT-CLOW)))/(2.*GP)
      C
      C       CALCULATE LAYERED CLOUD COVER
              CLD1=CLOW
              CLD2=CLOW*REDUCE+(1.-CLOW*REDUCE)*G2*CLD
              CLD3=G3*CLD
              RETURN
      C
              ENTRY CASE4(G2,G3,CLOW,CTOT,CLD1,CLD2,CLD3,REDUCE)
      C
      C       ROUTINE TO CALCULATE THREE LAYERS OF CLOUD COVER GIVEN LOWEST
      C           CLOUD COVER AND TOTAL CLOUD COVER ASSUMING A CB IN LAYERS
      C           1, 2, AND 3 WITH A RANDOM LAYERS 2 AND 3.
      C
      C       G2 = PROBABILITY OF RANDOM CLOUD IN LAYER 2
      C       G3 = PROBABILITY OF RANDOM CLOUD IN LAYER 3
      C       CLOW = CLOUD COVER OF CB(RANGE 0 - 1)
      C       CTOT = TOTAL CLOUD COVER (RANGE 0 - 1)
      C       CLD1 = CLOUD COVER OF LAYER 1 (RANGE 0 - 1)
      C       CLD2 = CLOUD COVER OF LAYER 2 (RANGE 0 - 1)
      C       CLD3 = CLOUD COVER OF LAYER 3 (RANGE 0 - 1)
      C       REDUCE = CB REDUCTION FACTOR
      C
      C       CALCULATE INTERMEDIATE FACTORS
      C
              GS=(G2+G3)*(1.-CLOW)
              GP=G2*G3*(1.-CLOW)
              CLD=(GS-SQRT(GS**2-4.*GP*(CTOT-CLOW)))/(2.*GP)
      C
      C       CALCULATE LAYERED CLOUD COVER
      C
              CLD1=CLOW
              CLD2=CLOW*REDUCE+(1.-REDUCE*CLOW)*G2*CLD
              CLD3=CLOW*REDUCE**2+(1.-CLOW*REDUCE**2)*G3*CLD
              RETURN
      C
```

CFAS SUBPROGRAM ELEMENT CASES

```
              ENTRY CASE5(G2,G3,CLOW,CTOT,CLD1,CLD2,CLD3)
        C
        C     ROUTINE TO CALCULATE THREE LAYERS OF CLOUD COVER GIVEN LOWEST
        C          CLOUD COVER AND TOTAL CLOUD COVER ASSUMING LAYERS ARE
        C          COMPLETELY RANDOM.
        C
        C     G2 = PROBABILITY OF RANDOM CLOUD IN LAYER 2
        C     G3 = PROBABILITY OF RANDOM CLOUD IN LAYER 3
        C     CLOW = LOWEST CLOUD COVER (RANGE 0 - 1)
        C     CTOT = TOTAL CLOUD COVER (RANGE 0 - 1)
        C     CLD1 = CLOUD COVER OF LAYER 1 (RANGE 0 - 1)
        C     CLD2 = CLOUD COVER OF LAYER 2 (RANGE 0 - 1)
        C     CLD3 = CLOUD COVER OF LAYER 3 (RANGE 0 - 1)
        C
        C     CALCULATE INTERMEDIATE FACTORS.
        C
              GS=(G2+G3)*(1.-CLOW)
              GP=G2*G3*(1.-CLOW)
              CLD=(GS-SQRT(GS**2-4.*GP*(CTOT-CLOW)))/(2.*GP)
        C
        C     CALCULATE LAYERED CLOUD COVER
        C
              CLD1=CLOW
              CLD2=G2*CLD
              CLD3=G3*CLD
              RETURN
        C
              ENTRY CASE6(CLOW,CTOT,CLD1,CLD2)
        C
        C     ROUTINE TO CALCULATE TWO LAYERS OF CLOUD COVER GIVEN LOWEST
        C     CLOUD COVER AND TOTAL CLOUD COVER ASSUMING LAYERS ARE
        C        COMPLETELY RANDOM.
        C     CLOW = LOWEST CLOUD COVER (RANGE 0 - 1)
        C     CTOT = TOTAL CLOUD COVER (RANGE 0 - 1)
        C     CLD1 = CLOUD COVER OF LAYER 1 (RANGE 0 - 1 )
        C     CLD2 = CLOUD COVER OF LAYER 2 (RANGE 0 - 1 )

        C     CALCULATE LAYERED CLOUD COVER

              CLD1=CLOW
              CLD2=(CTOT-CLOW)/(1.-CLOW)
              RETURN
              END
```

@HDG,P CFAS SUBPROGRAM ELEMENT CFEXEC

@PRT,S CFAS.CFEXEC
FURPUR 0026-10/28-13:57

CFAS SUBPROGRAM ELEMENT CFEXEC

CLOUD-FOG*CFAS.CFEXEC

```
          SUBROUTINE CFEXEC(TASK,TIME,OBSRPT,XO,YO,XLN,YLN,LAST,TYMOLD,DSP,
         *DIST,TYMC,ISSQ,NSSQ,NBKOUT,IDENT)
C
C         THIS ROUTINE IS THE INTERFACE BETWEEN THE EXPERIMENTAL PROTOTYPE
C         AUTOMATIC METEOROLOGICAL SYSTEM (EPAMS) AND THE CLOUD-FOG ANALYSIS
C         SYSTEM (CFAS). IN ADDITION CFEXEC DIRECTS THE INTERPRETATION OF
C         SURFACE AND UPPER AIR OBSERVATIONS AND REPORTS (OBS/REP) AND THE
C         CREATION OR UPDATES OF THE CLOUD FOG DATA BASE (CFDB).

C
C         INPUT DATA (FORMAL PARAMETERS)

C         TASK = TASK REQUESTED BY EPAMS
C                1 = SET UP THE OBS/REP STORAGE FILES
C                2 = INPUT OBS/REP
C                3 = CREATE A NEW CFDB
C                4 = UPDATE THE LATEST CFDB ON FILE
C         TIME = REFERENCE TIME OF CFDB CREATION OR UPDATE
C         OBSRPT = OBS/REP
C         XO = DISTANCE EAST FROM XREF OF THE LOWER LEFT HAND CORNER OF THE
C              SUB-WINDOW IN THE CFDB TO BE UPDATED, KM.
C         YO = DISTANCE NORTH FROM YREF OF THE LOWER LEFT HAND CORNER OF THE
C              SUB-WINDOW IN THE CFDB TO BE UPDATED, KM.
C         XLN = EAST-WEST LENGTH OF UPDATED SUB-WINDOW, KM.
C         YLN = NORTH-SOUTH LENGTH OF UPDATED SUB-WINDOW, KM.
C         LAST = SEQUENCE NUMBER OF THE LAST OBS/REP STORED.
C         TYMOLD = TIME OF OLDEST OBS/REP TO BE USED IN A CREATION OR UPDATE
C         DSP = MAXIMUM DISTANCE BETWEEN OBS/REP TO BE COMBINED INTO A
C               BEST REPORT, KM.
C         DIST = DISTANCE CONSTANTS IN WEIGHTING FUNCTION, KM.
C            DIST(1) USED WHEN CONVECTIVE CLOUDS ONLY PRESENT.
C            DIST(2) USED WHEN CONVECTIVE AND MIDDLE CLOUDS ONLY ARE
C                    PRESENT OR WHEN SHOWERY TYPE PRCIPITATION PRESENT OR
C                    PAST WEATHER.
C            DIST(3) USED FOR ALL OTHER CASES.
C         TYMC = TIME CONSTANTS IN WEIGHTING FUNCTION, MINUTES.
C            TYMC(1) USED WHEN CONVECTIVE CLOUDS ONLY PRESENT.
C            TYMC(2) USED WHEN CONVECTIVE AND MIDDLE CLOUDS ONLY ARE
C                    PRESENT OR WHEN SHOWERY TYPE PRCIPITATION PRESENT OR
C                    PAST WEATHER.
C            TYMC(3) USED FOR ALL OTHER CASES.
C         ISSQ = SEARCH SQUARE SIZES, NO. OF GRID POINTS.
C         NSSQ = NO. OF SEARCH SQUARES USED IN ANALYSIS.
C         NBKOUT = BLOCK NO. IN THE CFDB FILE TO WHICH THE CREATION OR
C                  UPDATE IS TO BE TRANSFERRED.
C         IDENT = TEN WORDS OF USER SUPPLIED IDENTIFICATION INFORMATION THAT
C                 PRECEEDS THE CLOUD-FOG-WEATHER DATA ON THE FILE.

C         DATA STATEMENTS
C
C         XREF = EAST-WEST UTM GRID COORDINATE OF THE LOWER LEFT HAND CORNER
C                OF THE CFDB WINDOW, KM.
C         YREF = NORTH-SOUTH UTM GRID COORDINATE OF THE LOWER LEFTHAND CORNER
C                OF THE CFDB WINDOW, KM.
C         CMRD = CENTRAL MERIDIAN OF THE WINDOW, DEGREES (+ IF WESTERN HEMI-
C                SPHERE, - IF ESTERN HEMISPHERE)
C         LSFILE = LOGICAL DEVICE NO. OF TEMPORARY STORAGE FILE USED IN
C                  'COMOBR'.
```

```
C      NCFF = LOGICAL SYSTEM FILE NO. OF THE CFDB FILE.
C      ILPR = DEVICE NO. OF LINE PRINTER
C      ICPR = LOGICAL DEVICE NO. OF CONSOLE PRINTER.
C      GRDPH = CFDB GRID POINT HEIGHT, METERS.
C      MNBR = MINIMUM NUMBER OF BEST REPORTS REQUIRED TO CALCULATE CFDB
C              PARAMETERS AT GRID POINT.

C      PARAMETERS
C
C      GRD = CFDB GRID. (GRID POINT SPACING, KM.)
C      LNTHX = EAST-WEST LENGTH OF THE CFDB WINDOW, KM.
C      LNTHY = NORTH-SOUTH LENGTH OF THE CFDB WINDOW, KM.
C      NOBR = MAXIMUM NUMBER OF OBS/REP THAT CAN BE USED IN A CREATION
C             OR UPDATE.

C      OBS/REP INPUT ELEMENTS

C      IX = X DISTANCE OF OBS/REP SITE FROM IXREF, HECTOMETERS.
C      IY = Y DISTANCE OF OBS/REP SITE FROM IYREF, HECTOMETERS.
C      IZ = OBS/REP SITE ELEVATION ABOVE MEAN SEA LEVEL, METERS.
C      ITIME = TIME OF OBS/REP (0-1440)
C      ITYPE = TYPE OF OBS/REP
C                 1 = AIRWAYS, -1 IF A SPECIAL.
C                 2 = METAR, -2 IF A SPECIAL (SPECI)
C                 3 = SYNOP
C                 4 = UPPER AIR (RAOB), -4 IF A SPECIAL
C                 5 = AFGWC 3D-NEPH OUTPUT
C      FOR EXPLANATION OF REMAINING OBS/REP INPUT ELEMENTS CONSULT
C      LISTINGS OF SUBROUTINE SFDINT IF A SURFACE OBS/REP OR
C      SUBROUTINE UADINT IF AN UPPER AIR OBS/REP.

C      CFDB PARAMETERS DETERMINED FROM OBS/REP.

C      IOBC = SEQUENCE NO. OF OBS/REP.
C      IVALU = CFDB INFORMATION VALUE OF THE OBS/REP
C      NTCLC = TOTAL CLOUD COVER, (00 TO 100)
C      NCEIL = HEIGHT OF CEILING LAYER (AGL), DEKAMETERS. MINUS IF A
C               VARIABLE CEILING. LAST DIGIT OF NCEIL INDICATES THE
C               METHOD BY WHICH THE CEILING WAS DETERMINED.
C                   1 = MEASURED
C                   2 = AIRCRAFT
C                   3 = BALLOON
C                   4 = RADAR
C                   5 = ESTIMATED
C                   6 = INDEFINITE
C      NVV = PREVAILING SURFACE VISIBILITY, METERS. MINUS IF VARIABLE.
C      MINBAS = HEIGHT OF BASE OF LOWEST CLOUD, DEKAMETERS.
C      MAXTOP = HEIGHT OF TOP OF HIGHEST CLOUD THAT COULD BE DETERMINED
C                FROM OBS/REP ELEMENTS, DEKAMETERS.
C      MSPWE = MOST SIGNIFICANT PRESENT WEATHER ELEMENT. (WMO CODE 4677)
C      LCOV(I) = PERCENT CLOUD COVER IN THE CFDB LAYERS, (00 TO 100).
C                                 CFDB LAYERS
C                     LAYER          BOTTOM                      TOP
C                       1         0 FEET       0 METERS    150 FEET     45 METERS
C                       2       150           45           300          91
C                       3       300           91           600         183
C                       4       600          183          1000         305
C                       5      1000          305          2000         610
C                       6      2000          610          3500        1067
```

CFAS SUBPROGRAM ELEMENT CFEXEC

```
C                                  7    3500      1067        5000       1524
C                                  8    5000      1524        6500       1981
C                                  9    6500      1981       10000       3048


      INTEGER TASK,TIME,OBSRPT,SKYCOV,CEILNG,CLDBAS,CLDTOP,WEATHR,VISIB,
     *GRDPH,GRDPV,TYMOLD,CFASD

      PARAMETER GRD=25,LNTHX=200,LNTHY=200,IP=LNTHX/GRD,JP=LNTHY/GRD,
     * ICFDB=10+(IP*JP*15), IJP=IP*JP
      PARAMETER NOBR=600

      COMMON /MAP/XREF,YREF,CMRD,LNX,LNY,GRDPS,GRDPH(IP,JP)
      COMMON /OBSREP/IX,IY,IZ,ITIME,IOBC,ITYPE,IVALU,NTCLC,NCEIL,NVV,
     *MINBAS,MAXTOP,MSPWE,LCOV(9),ICL,ITSC,ICM,ICH,ICTS(10),NWEA(7),IPW,
     *NOUSE(99)
      COMMON /INTOBR/INOBS(23,NOBR)
      COMMON /CFDB/ JDENT(10),SKYCOV(IP,JP),CEILNG(IP,JP),VISIB(IP,JP),
     *CLDBAS(IP,JP),CLDTOP(IP,JP),WEATHR(IP,JP),LAYCOV(IP,JP,9)
      COMMON /OUTPT/IBEG,IEND,JBEG,JEND
      DIMENSION OBSRPT(143),GRDPV(IP,JP,15),INOBEL(44),KOBR(143),DIST(3)
     *,TYMC(3),ISSQ(5),LCOVA(9),LCOVB(9),CFASD(ICFDB),IDENT(10)

      EQUIVALENCE (KOBR(1),IX),(CFASD(11),GRDPV(1,1,1),SKYCOV(1,1))

      DATA XREF/-1500./,YREF/3900./,CMRD/90./
      DATA GRDPH/IJP*0/
      DATA LSFILE/3/,NCFF/4/
      DATA ILPR/5/
      DATA MISS/-32768/
      DATA MNBR/1/

      GO TO (10,20,70,70),TASK
C
C     COME HERE TO INITIALIZE AND SET UP OBS/REP STORAGE FILES
C
   10 CALL BEGIN
      IOBC=0
      RETURN
C
C     COME HERE TO INTERPRET AND FILE AN OBS/REP.
C
   20 DO 25 K=1,143
   25 KOBR(K)=OBSRPT(K)
      IOBC=LAST+1
      NTR=IABS(ITYPE)
      GO TO (30,30,30,40,50),NTR
C
C     COME HERE TO INTERPRET A SURFACE OBS/REP
C
   30 CALL SFDINT
      GO TO 60
C
C     COME HERE TO INTERPRET AN UPPER AIR OBS/REP.
C
   40 CALL UADINT
      GO TO 60
C
C     COME HERE TO PROCESS CLOUD-FOG DATA FROM THE AFGWC 3D-NEPH OUTPUT
```

CFAS SUBPROGRAM ELEMENT CFEXEC

```
      C
         50 CALL AFDINT
         60 DO 65 K=1,44
         65 OBSRPT(K)=KOBR(K)
            CALL STOREC(OBSRPT)
            LAST=IOBC
            IF(LAST .EQ. NOBR) LAST=0
            RETURN
         70 NOB=0
            DO 80 K=1,10
         80 JDENT(K)=IDENT(K)
            INCODE=1
      C
      C     INSURE THAT TYMOLD IS NOT MORE THAN 720 MINUTES (12 HOURS) PRIOR
      C        TO TIME. RESET TYMOLD TO TIME-720 IF NECESSARY.
      C
            IF(TIME .GT. TYMOLD) GO TO 90
            ITEMP=TYMOLD
            TYMOLD=TYMOLD-1440
         90 IDIF=TIME-TYMOLD
            IF(TYMOLD .LT. 0) TYMOLD=ITEMP
            IF(IDIF .LE. 720) GO TO 100
            TYMOLD=TIME-720
            IF(TYMOLD .LT. 0) TYMOLD=1440+TYMOLD
      C
            IPRT=ILPR
            WRITE(IPRT,2000) IDIF,TYMOLD
       2000 FORMAT(' TIME DIFFERENCE BETWEEN REFERENCE TIME AND TIME OF OLDEST
           * USEABLE OBS/REP = ',I3,' MINUTES'/' TIME OF OLDEST USEABLE OBS/RE
           *P RESET TO ',I4,' MINUTES WHICH IS 720 MINUTES PRIOR TO REFERENCE
           *TIME'//)
      C
      C     RETREIVE OBS/REP IN REVERSE CHRONOLOGICAL ORDER FROM TIME TO
      C        TYMOLD
      C
        100 CALL RETOBR(INCODE,TIME,INOBEL,NOMORE,TYMOLD)
            INCODE=2
      C
      C     JUMP TO 120 IF THERE ARE NO MORE OBS/REP IN THE DATA BASE.
      C
            IF(NOMORE .EQ. 1) GO TO 120
            NOB=NOB+1
            DO 110 NEL=1,23
        110 INOBS(NEL,NOB)=INOBEL(NEL)
      C
      C     JUMP BACK TO 100 AND ATTEMPT TO RETRIEVE ANOTHER OBS/REP IF THE
      C     MAXIMUM USEABLE NUMBER HAS NOT BEEN REACHED.
      C
            IF(NOB .LT. NOBR) GO TO 100
      C
      C     DETERMINE THE LOWEST ALTITUDE IN THE LIST OF OBS/REP AND GRID
      C     POINTS
      C
        120 IHREF=32000
            DO 130 N=1,NOB
        130 IHREF=MIN0(IHREF,INOBS(3,N))
            DO 140 I=1,IP
            DO 140 J=1,JP
        140 IHREF=MIN0(IHREF,GRDPH(I,J))
```

CFAS SUBPROGRAM ELEMENT CFEXEC

```
      C
      C      REFERENCE CEILING, MINIMUM BASE OF CLOUD, MAXIMUM TOP OF CLOUDS,
      C      AND THE CFDB LAYERS TO THE REFERENCE ALTITUDE, IHREF.
      C
             DO 190 N=1,NOB
             DO 160 M=9,13
             IF(INOBS(M,N) .EQ. MISS) GO TO 160
             MGT=M-8
             GO TO (145,150,150,150,160),MGT
         145 ISYN=2
             IF(INOBS(9,N) .LT. 0) ISYN=1
             MTMP=IABS(INOBS(9,N))
             MDS=MOD(MTMP,10)
             MTMP=MTMP/10
             INOBS(9,N)=((MTMP*10)+INOBS(3,N)-IHREF)/10
             IF(INOBS(9,N) .LT. 0) INOBS(9,N)=0
             INOBS(9,N)=((10*INOBS(9,N))+MDS)*((-1)**ISYN)
             GO TO 160
         150 INOBS(M,N)=((10*INOBS(M,N))+INOBS(3,N)-IHREF)/10
             IF(INOBS(M,N) .LT. 0) INOBS(M,N)=0
         160 CONTINUE
             DO 170 M=14,22
             MGT=M-13
             LCOVB(MGT)=INOBS(M,N)
         170 LCOVA(MGT)=MISS
             IHB=INOBS(3,N)
             CALL MVLCOV(LCOVA,LCOVB,IHREF,IHB)
             DO 180 M=14,22
             MGT=M-13
         180 INOBS(M,N)=LCOVA(MGT)
         190 CONTINUE

      C      RANK OBS/REP WITHIN 'DSP' KM. OF A GIVEN OBS/REP. RESOLVE CONFLICT
      C      ING INFORMATION IN THE SAME CFDB ELEMENTS OF THE SEVERAL OBS/REP
      C      ON THE BASIS OF RANK AND COMBINE NON CONFLICTING INFORMATION INTO
      C      A BEST OBS/REP AT THE SITE OF THE GIVEN OBSREP.

             CALL COMOBR(NOB,DSP,TIME,LSFILE)
             MGT=TASK-2
             GO TO(200,210),MGT
      C
      C      COME HERE TO CREATE A NEW CFDB
      C
         200 IBEG=1
             IEND=IP
             JBEG=1
             JEND=JP
             GO TO 220

      C      COME HERE TO UPDATE AN EXISTING CLOUD FOG DATA-BASE.

         210 IXO=XO
             IBEG=IXO/GRD+1
             IXO=XO+XLN
             IEND=IXO/GRD+1
             IXO=MOD(IXO,GRD)
             IF(IXO .GT. 0) IEND=IEND+1
             JYO=YO
             JBEG=JYO/GRD+1
```

```
CFAS SUBPROGRAM ELEMENT CFEXEC

          JYO=YO+YLN
          JEND=JYO/GRD+1
          JYO=MOD(JYO,GRD)
          IF(JYO .GT. 0) JEND=JEND+1
          IF(IEND .GT. IP) IEND=IP
          IF(JEND .GT. JP) JEND=JP
      220 CALL CFMAP(IBEG,IEND,JBEG,JEND,DIST,TYMC,ISSQ,NSSQ,MNBR,TIME,NOB)

C         REFERENCE CREATED OR UPDATED CFDB PARAMETERS TO GROUND LEVEL.

          DO 280 I=IBEG,IEND
          DO 280 J=JBEG,JEND
          DO 250 M=1,6
          IF(GRDPV(I,J,M) .EQ. MISS) GO TO 250
          GO TO (250,230,250,240,240,250),M
      230 ISYN=2
          IF(GRDPV(I,J,2) .LT. 0) ISYN=1
          MTMP=IABS(GRDPV(I,J,2))
          GRDPV(I,J,2)=((MTMP*10)+IHREF-GRDPH(I,J))/10
          IF(GRDPV(I,J,2) .LT. 0) GRDPV(I,J,2)=0
          GRDPV(I,J,2)=GRDPV(I,J,2)*((-1)**ISYN)
          GO TO 250
      240 GRDPV(I,J,M)=((GRDPV(I,J,M)*10)+IHREF-GRDPH(I,J))/10
          IF(GRDPV(I,J,M) .LT. 0) GRDPV(I,J,M)=0
      250 CONTINUE
          DO 260 M=7,15
          MY=M-6
          LCOVB(MY)=GRDPV(I,J,M)
      260 LCOVA(MY)=MISS
          IHA=GRDPH(I,J)
          CALL MVLCOV(LCOVA,LCOVB,IHA,IHREF)
          DO 270 M=7,15
          MY=M-6
          MTMP=MOD(LCOVA(MY),5)
          IF(MTMP .NE. 1) GO TO 270
          LCOVA(MY)=-(LCOVA(MY)-1)
      270 GRDPV(I,J,M)=LCOVA(MY)

C         INSURE MINBAS LESS THAN MAXTOP.

          IF(GRDPV(I,J,5) .LT. GRDPV(I,J,4)) GRDPV(I,J,5)=GRDPV(I,J,4)

C         INSURE THAT TOTAL SKY COVER NOT LESS THAN THE CLOUD COVER IN ANY
C         LAYER.

          DO 275 M=7,15
          IF(GRDPV(I,J,M) .EQ. MISS) GO TO 275
          MTMP=IABS(GRDPV(I,J,M))
          GRDPV(I,J,1)=MAX0(GRDPV(I,J,1),MTMP)
      275 CONTINUE
      280 CONTINUE

C         OUTPUT CREATED OR UPDATED CFDB.

          NWDBLK=ICFDB
          CALL BLKOUT(NWDBLK,CFASD,NBKOUT,NCFF,ISTAT)
          RETURN
          END
```

I-16

```
CFAS SUBPROGRAM ELEMENT CFEXEC

@HDG.P CFAS SUBPROGRAM ELEMENT CFLAY

@PRT.S CFAS.CFLAY
FURPUR 0026-10/28-13:57
```

CFAS SUBPROGRAM ELEMENT CFLAY

```
CLOUD-FOG*CFAS.CFLAY
              SUBROUTINE CFLAY(NBASE,NTOP,MINLAY,MAXLAY)
        C
        C     ROUTINE TO FIND MINIMUM AND MAXIMUM CFDB LAYERS INFLUENCED BY
        C     CLOUD LAYERS CONSTRUCTED FROM OBS/REP. 0 IS RETURNED IF NO CFDB
        C     LAYERS ARE AFFECTED.
        C
        C     NBASE = BASE IN FEET ABOVE TERRAIN.
        C     NTOP = TOP IN FEET ABOVE TERRAIN.
        C     MINLAY = MINIMUM LAYER ABOVE TERRAIN.
        C     MAXLAY = MAXIMUM LAYER ABOVE TERRAIN    .
        C
              DIMENSION LEVEL(10)

              DATA LEVEL/0,150,300,600,1000,2000,3500,5000,6500,10000/

        C     ROUND BASE AND TOP TO NEAREST 100 FEET.

              IBASE=((NBASE+50)/100)*100
              ITOP=((NTOP+50)/100)*100

        C     RETURN 0 IF BASE ABOVE MAXIMUM LAYER.

              IF(IBASE .LT. LEVEL(10)) GO TO 50
              MINLAY=0
              MAXLAY=0
              RETURN

        C     RETURN 0 IF TOP BELOW MINIMUM LAYER.

           50 IF(ITOP .GT. LEVEL(1)) GO TO 60
              MINLAY=0
              MAXLAY=0
              RETURN

        C     FIND MINIMUM LAYER ABOVE TERRAIN

           60 DO 70 IL=1,10
              LEV=11-IL
              IF(IBASE .GE. LEVEL(LEV)) GO TO 70
              MINLAY=LEV-1
           70 CONTINUE

        C     FIND MAXIMUM LAYER ABOVE TERRAIN.

              DO 80 LEV=1,9
              IF(ITOP .LT. LEVEL(LEV)) GO TO 80
              MAXLAY=LEV
          80  CONTINUE
              RETURN
              END
```

```
@HDG,P CFAS SUBPROGRAM ELEMENT CFMAIN

@PRT,S CFAS.CFMAIN
FURPUR 0026-10/28-13:57
```

CFAS MAIN PROGRAM ELEMENT CFMAIN

CLOUD-FOG*CFAS.CFMAIN

```
      C     CFMAIN
      C     TEST DRIVER FOR THE CFAS
            PARAMETER GRD=25,LNTHX=200,LNTHY=200,IG=LNTHX/GRD,JG=LNTHY/GRD,
           * ICFDD=10+(IG*JG*15)
            INTEGER TASK,TIME,TYMOLD,CFASD,GRDPV
            COMMON /IDAT/ JX,JY,JZ,ITIME,IOBS,ITYPE,IVALU,NTCLC,NCEIL,NVV,
           *MINBAS,MAXTOP,MSPWE,LCOV(9),ICL,ITSC,ICM,ICH,ICTS(10),NWEA(7),IPW,
           *IWD,IWS,IPPP,ITT,ITD,IVIS,NH,IH,NS(10),IHS(10),ITHN(10),ICLG,ICLGV
           *,IVISC,NOUSE(56)
            COMMON /MAP/ XREF,YREF,CMRD
            COMMON /OUTPT/IBEG,IEND,JBEG,JEND
            DIMENSION IDAT(143),JDAT(143),IZ(30),IP(30),IT(30),IDD(30),DIST(3)
           *,TYMC(3),ISSQ(5),CFASD(ICFDD),GRDPV(IG,JG,15),IDENT(10),Z(30)
           *,P(30),T(30),DD(30)
            EQUIVALENCE (IDAT(1),JX),(IDAT(23),ICL,IZ(1)),(IDAT(53),NS(1),IP(1
           *)),(IDAT(83),ICLG,IT(1)),(IDAT(113),IDD(1)),(IDAT(143),NRRL),
           *(CFASD(11),GRDPV(1,1,1)),(CFASD(1),IDENT(1))
            DATA MISS/-32768/
            DATA NCFF/4/
            DATA XREF/-1500./,YREF/3900./,CMRD/90./
            LAST=0
            NPRT=0
            NBKOUT=0
          5 READ  (5,1000) TASK,NPRT,TIME
       1000 FORMAT(8I10)
            WRITE (6,2000)TASK,NPRT,TIME
            MFG=1
       2000 FORMAT('1',5X,'TASK= ',I2,5X,'NPRT=',I4,5X,'TIME=',I5)
            GO TO (130,10,200,200),TASK
         10 DO 20 I=1,143
         20 IDAT(I)=MISS
            READ (5,1000),JX,JY,JZ,ITIME,ITYPE,IVIS,NC
            MFG=ITYPE/10
            ITYPE=MOD(ITYPE,10)
            MT=IABS(ITYPE)
            GO TO (30,30,30,80,100),MT
         30 IF(NPRT .EQ. 0) GO TO 33
            WRITE (6,2005)
       2005 FORMAT(/)
            WRITE (6,2010)JX,JY,JZ,ITIME,ITYPE,IVIS,NC
       2010 FORMAT(4X,'JX',8X,'JY',8X,'JZ',6X,'ITIME',5X,'ITYPE',6X,'IVIS',6X,
           *'NC'/7(2X,I6,2X)/)
         33 GO TO (35,50,40,80,100),MT
         35 READ (5,1000) ICLG,ICLGV,IVISC
            IF(NPRT .EQ. 0) GO TO 50
            WRITE(6,2020) ICLG,ICLGV,IVISC
       2020 FORMAT(3X,'ICLG',5X,'ICLGV',5X,'IVISC'/3(2X,I6,2X)/)
            GO TO 50
         40 READ (5,1000) ITSC,NH,ICL,IH,ICM,ICH,IPW
            IF(NPRT .EQ. 0) GO TO 50
            WRITE(6,2030) ITSC,NH,ICL,IH,ICM,ICH,IPW
       2030 FORMAT(3X,'ITSC',7X,'NH',7X,'ICL',8X,'IH',7X,'ICM',7X,'ICH',7X,'IP
           *W'/7(2X,I6,2X)/)
         50 READ (5,1000) (NWEA(I),I=1,7)
            IF(NPRT .EQ. 0) GO TO 55
            WRITE (6,2035) (NWEA(I),I=1,7)
       2035 FORMAT(2X,'NWEA(1)',3X,'NWEA(2)',3X,'NWEA(3)',3X,'NWEA(4)',3X,'NWE
           *A(5)',3X,'NWEA(6)',3X,'NWEA(7)'/7(3X,I4,3X)/)
```

I-17

CFAS MAIN PROGRAM ELEMENT CFMAIN

```
         55 IF(NC .EQ. 0) GO TO 110
            READ  (5,1010) (NS(I),ICTS(I),IHS(I),ITHN(I),I=1,NC)
       1010 FORMAT(4I10)
            IF(NPRT .EQ. 0) GO TO 110
            WRITE(6,2040)
       2040 FORMAT(4X,'NS',9X,'ICTS',8X,'IHS',9X,'ITHN'/)
            WRITE(6,2050)(NS(I),ICTS(I),IHS(I),ITHN(I),I=1,NC)
       2050 FORMAT(4(2X,I6,4X))
            GO TO 110
         80 IF(NPRT .EQ. 0) GO TO 85
            WRITE (6,2005)
            WRITE(6,2060) JX,JY,JZ,ITIME,ITYPE
       2060 FORMAT(4X,'JX',8X,'JY',8X,'JZ',6X,'ITIME',5X,'ITYPE'/5(2X,I6,2X)/)
         85 I=0
         90 I=I+1
            READ  (5,1000) IZ(I),IP(I),IT(I),IDD(I)
            Z(I)=FLOAT(IZ(I))*10.
            P(I)=FLOAT(IP(I))*.1
            P(I)=ABS(P(I))
            T(I)=FLOAT(IT(I))*.1
            DD(I)=FLOAT(IDD(I))*.1
            IF((IP(I) .GE. 0) .OR. (IP(I) .EQ. MISS)) GO TO 90
            NRRL=I
            IF(NPRT .EQ. 0) GO TO 110
            WRITE (6,2070)
       2070 FORMAT(9X,'IZ',10X,'IP',10X,'IT',10X,'IDD',10X,'Z',12X,'P',11X,'T'
           *11X,'DD'/)
            WRITE(6,2080) (IZ(I),IP(I),IT(I),IDD(I),Z(I),P(I),T(I),DD(I),I=1,
           *NRRL)
       2080 FORMAT(3X,4I12,4F12.2)
            GO TO 110
        100 IF(NPRT .EQ. 0) GO TO 105
            WRITE (6,2005)
            WRITE (6,2060) JX,JY,JZ,ITIME,ITYPE
        105 READ  (5,1000) NTCLC,NCEIL,NVV,MINBAS,MAXTOP,MSPWE,(LCOV(I),I=1,9)
            IF(NPRT .EQ. 0) GO TO 110
            WRITE (6,2090) NTCLC,NCEIL,NVV,MINBAS,MAXTOP,MSPWE,(LCOV(I),I=1,9)
       2090 FORMAT(3X,'NTCLC',5X,'NCEIL',6X,'NVV',5X,'MINBAS',4X,'MAXTOP',4X,'
           *MSPWE',4X,'LCOV(1)',3X,'LCOV(2)'/8(2X,I6,2X)//2X,'LCOV(3)',3X,'LCO
           *V(4)',3X,'LCOV(5)',3X,'LCOV(6)',3X,'LCOV(7)',3X,'LCOV(8)',3X,'LCOV
           *(9)'/7(2X,I6,2X)///)
        110 DO 120 J=1,143
        120 JDAT(J)=IDAT(J)
        130 CALL CFEXEC(TASK,TIME,JDAT,XO,YO,XLN,YLN,LAST,TYMOLD,DSP,DIST,TYMC
           *,ISSQ,NSSQ,NBKOUT,IDENT)
            IF(MFG) 10,10,5
        200 READ  (5,1000) TIME,TYMOLD,NSSQ,(ISSQ(I),I=1,NSSQ)
            WRITE (6,2100) TIME,TYMOLD,NSSQ,(ISSQ(I),I=1,NSSQ)
       2100 FORMAT(/3X,'TIME',5X,'TYMOLD',5X,'NSSQ',4X,'ISSQ(1)',3X,'ISSQ(2)',
           *3X,'ISSQ(3)',3X,'ISSQ(4)'/8(2X,I6,2X)/)
            READ  (5,1020) DSP,(DIST(I),I=1,3),(TYMC(I),I=1,3)
       1020 FORMAT(8F10.1)
            WRITE (6,2110) DSP,(DIST(I),I=1,3),(TYMC(I),I=1,3)
       2110 FORMAT(/5X,'DSP',6X,'DIST(1)',5X,'DIST(2)',5X,'DIST(3)',5X,'TYMC(1
           *)',5X,'TYMC(2)',5X,'TYMC(3)'/7(2X,F8.1,2X)/)
            IF(TASK .EQ. 3) GO TO 250
            READ  (5,1020) XO,YO,XLN,YLN
            WRITE (6,2120) XO,YO,XLN,YLN
       2120 FORMAT(/5X,'XO',10X,'YO',9X,'XLN',9X,'YLN'/4(2X,F8.1,2X))
```

CFAS MAIN PROGRAM ELEMENT CFMAIN

```
  250 NBKOUT=NBKOUT+1
      CALL CFEXEC(TASK,TIME,JDAT,XO,YO,XLN,YLN,LAST,TYMOLD,DSP,DIST,TYMC
     *,ISSQ,NSSQ,NBKOUT,IDENT)
      NWDBLK=ICFDB
      CALL BLKIN(NWDBLK,CFASD,NBKOUT,NCFF,ISTAT)
      WRITE (6,2130) NBKOUT
 2130 FORMAT('1',9X,'CONTENTS OF BLOCK NO.',I3,' OF THE CFDB FILE'///)
      WRITE (6,2140) (CFASD(I),I=1,10)
 2140 FORMAT(4X,'IDENT= ',10(I6,2X)//4X,'GRID POINT DATA FOLLOWS'/)
      LYNC=10
  260 WRITE (6,2150)
 2150 FORMAT(2X,'I',4X,'J',4X,'SKYC',4X,'CEIL',4X,'VIS',5X,'BASE',4X,'TO
     *P',5X,'WETHR',3X,'LAY1',4X,'LAY2',4X,'LAY3',4X,'LAY4',4X,'LAY5',4X
     *,'LAY6',4X,'LAY7',4X,'LAY8',4X,'LAY9'/)
      DO 280 I=IBEG,IEND
      DO 280 J=JBEG,JEND
      LYNC=LYNC+1
      IF(LYNC .LT. 54) GO TO 270
      LYNC=3
      WRITE (6,2160)
 2160 FORMAT('1')
      WRITE (6,2150)
  270 WRITE (6,2170) I,J,(GROPV(I,J,M),M=1,15)
 2170 FORMAT(1X,I2,3X,I2,4X,I3,4X,I6,2X,I6,2X,I6,2X,I6,4X,I2,3X,9(1X,I6,
     *1X))
  280 CONTINUE
      GO TO 5
      END
```

CFAS SUBPROGRAM ELEMENT CFMAP

CLOUD-FOG*CFAS.CFMAP

```
      SUBROUTINE CFMAP(IBEG,IEND,JBEG,JEND,DIST,TYMC,ISSQ,NSSQ,MNBR,
     *MTIME,NOB)

C     THIS ROUTINE USES THE BEST REPORTS GENERATED BY COMOBR TO DETER-
C     MINE THE CFDB PARAMETERS AT SPECIFIED GRID POINTS IN THE WINDOW.

C     INPUT DATA

C     IBEG = I INDEX OF LEFT HAND EDGE OF WINDOW OR SUB-WINDOW.
C     IEND = I INDEX OF RIGHT HAND EDGE OF WINDOW OR SUB-WINDOW.
C     JBEG = J INDEX OF BOTTOM EDGE OF WINDOW OR SUB-WINDOW.
C     JEND = J INDEX OF TOP EDGE OF WINDOW OR SUB-WINDOW.
C     DIST = DISTANCE CONSTANTS IN WEIGHTING FUNCTION, KM.
C     TYMC = TIME CONSTANTS IN WEIGHTING FUNCTION, MINUTES.
C     ISSQ = SEARCH SQUARE SIZES, NO. OF GRID POINTS.
C     NSSQ = NUMBER OF SEARCH SQUARES.
C     MNBR = MINIMUM NUMBER OF BEST REPORTS REQUIRED TO CALCULATE CFDB
C            PARAMETERS AT A GRID POINT.
C     MTIME = MAP TIME (0 - 1440).
C     NOB = NUMBER OF OBS/REP.

      INTEGER GRDPV

      PARAMETER GRD=25,LNTHX=200,LNTHY=200,IP=LNTHX/GRD,JP=LNTHY/GRD
      PARAMETER NOBR=600, MNP=300

      COMMON /INTOBR/ INOBS(23,NOBR)
      COMMON /CFDB/ NOUSZ(10),GRDPV(IP,JP,15)

      DIMENSION JPT(NOBR),ISSQ(5),DIST(3),TYMC(3),KNPT(MNP),NPT(NOBR)

      DATA MISS /-32768/


C     STEP THROUGH THE CFDB PARAMETERS
      DO 220 M=8,22
      MGT=M-7
      IF(MGT .GT. 7) MGT=7

C     SEARCH BEST REPORT OBS/REP AND GENERATE A POINTER TABLE TO THOSE
C     REPORTS HAVING INFORMATION ON THE CFDB PARAMETER BEING ANALYZED.

      NUM=0
      DO 20 N=1,NOB
      IF(INOBS(M,N) .EQ. MISS) GO TO 20
      NUM=NUM+1
      JPT(NUM)=N
   20 CONTINUE

C     STEP THROUGH GRID POINTS.

      DO 210 I=IBEG,IEND
      IXD=GRD*(I-1)*10
      DO 210 J=JBEG,JEND
      IYD=GRD*(J-1)*10
      NFD=0
      DO 23 NJ=1,NUM
   23 NPT(NJ)=JPT(NJ)
```

CFAS SUBPROGRAM ELEMENT CFMAP

```
            NMPT=NUM

C           STEP THROUGH SEARCH SQUARES OF INCREASING SIZE.

            NSQ=1
         25 IF(NMPT .LE. 0) GO TO 85
            JSD=GRD*ISSQ(NSQ)*10
            N=1
         30 NN=NPT(N)
            NXD=INOBS(1,NN)
            NYD=INOBS(2,NN)
            IXDF=IABS(IXD-NXD)
            IYDF=IABS(IYD-NYD)
            IF((IXDF .GT. JSD) .OR. (IYDF .GT. JSD)) GO TO 50
            NFD=NFD+1
            KNPT(NFD)=NN
            IF(N .EQ. NMPT) GO TO 60
            NMPT=NMPT-1
            DO 40 NR=N,NMPT
         40 NPT(NR)=NPT(NR+1)
            GO TO 30
         50 N=N+1
            IF(N-NMPT) 30,30,70

C           JUMP TO 90 IF THE MINIMUM NUMBER OF BEST REPORTS USEABLE AT THE
C           GRID POINT HAS BEEN FOUND.

         60 NMPT=NMPT-1
         70 IF(NFD .GE. MNBR) GO TO 90
            NSQ=NSQ+1

C           JUMP TO 80 IF THE LARGEST SEARCH SQUARE HAS BEEN EXCEEDED.

            IF(NSQ-NSSQ) 25,25,80

C           JUMP TO 90 IF AT LEAST ONE BEST REPORT LYING WITHIN THE LARGEST
C           SEARCH SQUARE HAS BEEN FOUND.

         80 IF(NFD .GT. 0) GO TO 90
         85 MP=M-7
            GRDPV(I,J,MP)=MISS
            GO TO 210
         90 ITS=0
            SMWF=0.
            SMWFO=0.
            DO 180 N=1,NFD
            NB=KNPT(N)
            DXS=(IXD-INOBS(1,NB))**2
            DYS=(IYD-INOBS(2,NB))**2
            DIS=SQRT(DXS+DYS)
            TD=MTIME-INOBS(4,NB)
            IF(TD .LT. 0) TD=1440+TD
            IF(INOBS(23,NB) .EQ. MISS) GO TO 100
            LT=IABS(INOBS(23,NB))
            LT=MOD(LT,10)
            IF(LT .GT. 3) LT=3
            TC=TYMC(LT)
            DC=DIST(LT)*10.
            GO TO 110
```

CFAS SUBPROGRAM ELEMENT CFMAP

```
        100 TC=TYMC(3)
            DC=DIST(3)*10.
        110 WF=INOBS(7,NB)*EXP(-((DIS/DC)**2)-((TD/TC)**2))
            GO TO (140,120,120,140,140,150,130),MGT
        120 IF(INOBS(M,NB) .LT. 0) ITG=ITG+1
            LGD=IABS(INOBS(M,NB))
            IF(M .NE. 9) GO TO 160
            LGD=LGD/10
            GO TO 160
        130 LGD=MOD(INOBS(M,NB),5)
            IF(LGD .EQ. 1) ITG=ITG+1
            LGD=INOBS(M,NB)-LGD
            GO TO 160
        140 LGD=INOBS(M,NB)
            GO TO 160
        150 IF(WF .LT. SMWF) GO TO 180
            SMWF=WF
            IGPV=INOBS(13,NB)
            GO TO 180
        160 OBS=LGD
            SMWF=SMWF+WF
            SMWFO=SMWFO+(WF*OBS)
        180 CONTINUE
            XITG=ITG
            XNFD=NFD
            FRAC=XITG/XNFD
            GPV=SMWFO/SMWF
            GO TO (190,190,190,190,190,200,185),MGT
        185 GPV=GPV+2.5
            IGPV=GPV/5
            IGPV=IGPV*5
            IF(IGPV .EQ. 0) GO TO 200
            IF(FRAC .LT. .5) GO TO 200
            IGPV=IGPV+1
            GO TO 200
        190 IGPV=GPV+.5
            GO TO (200,195,195,200,200),MGT
        195 IF(FRAC .GE. .5) IGPV=-IGPV
        200 MGZ=M-7
            GRDPV(I,J,MGZ)=IGPV
        210 CONTINUE
        220 CONTINUE
            RETURN
            END
```

@HDG,P CFAS SUBPROGRAM ELEMENT RETOBR

@PRT,S CFAS.RETOBR

FURPUR 0026-11/05-10:30

CFAS SUBPROGRAM ELEMENT COMOBR

CLOUD-FOG*CFAS.COMOBR

```
      SUBROUTINE COMOBR(NOB,DSP,TIME,LSFILE)
C     RANKS, RESOLVES CONFLICTING INFORMATION, AND COMBINES CFDB ELE-
C     MENTS OF PROXIMATE OBS/REPS: THEN INSURES INTERNAL CONSISTENCY OF
C     COMBINED OBS/REP

      PARAMETER NOBR=600

      INTEGER GRD,GPEL,TIME

      COMMON /MAP/XREF,YREF,CMRD,LNTHX,LNTHY,GRD
      COMMON /INTOBR/INOBS(23,NOBR)

      DIMENSION GPEL(24,10),MTEMP(24),NREC(11),DS(11)

      DATA NWDBLK/23/
      DATA MISS/-32768/

      DSPH=DSP*10.
      NBKOUT=0
      DO 300 N=1,NOB
      ICT=1
      DO 20 M=1,23
   20 GPEL(M,ICT)=INOBS(M,N)
      GPEL(24,ICT)=1

C     COLLECT THE CLOSEST 10 OR LESS OBS/REP TO THE SITE OF OBS/REP NO.
C     INOBS(5,N), CALLED THE BEST REPORT SITE, WHICH ARE NO MORE THAN
C     'DSP' KM. FROM THE BEST REPORT SITE.

      DO 35 NN=1,NOB
      IF(NN .EQ. N) GO TO 35
      DXS=(INOBS(1,N)-INOBS(1,NN))**2
      DYS=(INOBS(2,N)-INOBS(2,NN))**2
      DIST=SQRT(DXS+DYS)
      IF(DIST .GT. DSPH) GO TO 35
      ICT=ICT+1
      NREC(ICT)=NN
      DS(ICT)=DIST
      IF(ICT .LE. 2) GO TO 35
      JX=ICT
   22 IF(DS(JX) .GE. DS(JX-1)) GO TO 30
      DTM=DS(JX-1)
      NTM=NREC(JX-1)
      DS(JX-1)=DS(JX)
      NREC(JX-1)=NREC(JX)
      DS(JX)=DTM
      NREC(JX)=NTM
      JX=JX-1
      IF(JX .GT. 2) GO TO 22
   30 IF(ICT .GE. 11) ICT=10
   35 CONTINUE

C     JUMP TO 40 IF OTHER OBS/REP ARE WITHIN DSP KM. OF BEST REPORT SITE

      IF(ICT .GT. 1) GO TO 40
      DO 38 M=1,23
   38 MTEMP(M)=GPEL(M,ICT)
      GO TO 290
```

CFAS SUBPROGRAM ELEMENT COMOBR

```
   40 DO 45 IC=2,ICT
      NX=NREC(IC)
      DO 43 M=1,23
   43 GPEL(M,IC)=INOBS(M,NX)
      GPEL(24,IC)=IC
   45 CONTINUE
   50 DO 90 NR=2,ICT
      NRR=NR
C     RANK OBS/REP ON URGENCY THEN TYPE. SPECIALS OF ALL TYPES OUTRANK
C     NON SPECIALS OF SAME OR OTHER TYPE. TYPES RANKED AS FOLLOWS:
C           1    AIRWAYS
C           2    METAR
C           3    SYNOP
C           4    UPPER AIR
C           5    AFGWC-3DNEPH
   55 JTP1=GPEL(6,NRR-1)+5
      IF(GPEL(6,NRR-1) .LT. 0) JTP1=IABS(GPEL(6,NRR-1))
      JTP2=GPEL(6,NRR)+5
      IF(GPEL(6,NRR) .LT. 0) JTP2=IABS(GPEL(6,NRR))
      IF(JTP2 - JTP1) 80,60,90
C     RANK ON BASIS OF TIME OF RECEIPT OF OBS/REP
   60 ITD1=TIME-GPEL(4,NRR-1)
      IF(ITD1 .LT. 0) ITD1=ITD1+1440
      ITD2=TIME-GPEL(4,NRR)
      IF(ITD2 .LT. 0) ITD2=ITD2+1440
      IF(ITD2 - ITD1) 80,70,90
C     RANK ON BASIS OF VALUE OF OBS/REP
   70 IF(GPEL(7,NRR-1)-GPEL(7,NRR)) 80,75,90
C     RANK ON DISTANCE FROM BEST REPORT SITE.
   75 IF(GPEL(24,NRR-1) .LE. GPEL(24,NRR)) GO TO 90
   80 DO 85 MT=1,24
      MTEMP(MT)=GPEL(MT,NRR-1)
      GPEL(MT,NRR-1)=GPEL(MT,NRR)
   85 GPEL(MT,NRR)=MTEMP(MT)
      NRR=NRR-1
      IF(NRR .GE. 2) GO TO 55
   90 CONTINUE
C     CREATE A BEST REPORT AT THE SITE OF OBS/REP NO. INOBS(5,N).
C     ASSIGN THE LOCATION, STATION ELEVATION, TIME SEQUENCE NO. AND
C     TYPE OF OBS/REP NO. INOBS(5,N) TO THIS BEST REPORT.
      DO 100 M=1,6
  100 MTEMP(M)=INOBS(M,N)
C     INITIALLY ASSIGN THE CFDB PARAMETERS OF THE LOWEST RANKING OBS/REP
C     WITHIN DSP KM. OF THE BEST REPORT SITE TO THE BEST REPORT.
      KL1=10
      MVAL=GPEL(7,ICT)
      DO 170 M=8,22
```

CFAS SUBPROGRAM ELEMENT COMOBR

```
               MTEMP(M)=GPEL(M,ICT)

      C        JUMP TO 130 FOR ALL CFDB PARAMETERS EXCEPT THE CEILING

               IF(M .NE. 9) GO TO 130

      C        DETERMINE THE CODE NO. INDICATING THE METHOD BY WHICH THE CEILING
      C        WAS MEASURED.

               KL1=IABS(GPEL(9,ICT))
               KL1=MOD(KL1,10)

      C        STEP UPWARD THROUGH RANKED OBS/REP AND REPLACE CFDB PARAMETERS
      C        PREVIOSLY ASSIGNED TO BEST REPORT BY CORRESPONDING PARAMETERS IN
      C        OBS/REP OF HIGHER RANK LOCATED WITHIN DSP KM. OF BEST REPORT SITE.
      C        DO NOT MAKE THE REPLACEMENT IF THE CFDB PARAMETER IN THE HIGHER
      C        RANKING OBS/REP IS MISSING.

           130 DO 170 I=2,ICT
               IRV=ICT+1-I
               IF(GPEL(M,IRV) .EQ. MISS) GO TO 170

      C        JUMP TO 150 FOR ALL PARAMETERS EXCEPT CEILING

               IF(M .NE. 9) GO TO 150
               KL2=IABS(GPEL(M,IRV))
               KL2=MOD(KL2,10)

      C        DO NOT REPLACE CEILING UNLESS METHOD OF CEILING DETERMINATION IN
      C        HIGHER RANKING OBS/REP IS ALSO A HIGHER RANKING METHOD THAN WAS
      C        USED IN DETERMINING THE CEILING VALUE PRESENTLY ASSIGNED TO THE
      C        BEST REPORT.

               IF(KL1 .LE. KL2) GO TO 170
               KL1=KL2
           150 MTEMP(M)=GPEL(M,IRV)
           170 CONTINUE

      C        INSURE THAT TOTAL SKY COVER IS NOT LESS THAN THE PERCENT CLOUD
      C        COVER IN ANY LAYER.

               DO 220 M=14,22
           220 MTEMP(8)=MAX0(MTEMP(8),MTEMP(M))

      C        INSURE MINBAS LESS THAN MAXTOP

               IF(MTEMP(12) .LE. MTEMP(11)) MTEMP(12)=MISS
               DO 230 I=2,ICT
               IRV=ICT+1-I
               IF((IABS(GPEL(6,IRV)) .GT. 3) .OR. (GPEL(23,IRV) .EQ. MISS)) GO TO
              * 230
               MTEMP(23)=GPEL(23,IRV)
           230 MVAL=MVAL+GPEL(7,IRV)
               MVAL=MVAL/ICT
               MTEMP(7)=((2*INOBS(7,N))+MVAL)/3
           290 NBKOUT=NBKOUT+1
               CALL BLKOUT(NWDBLK,MTEMP,NBKOUT,LSFILE,ISTAT)
           300 CONTINUE
```

CFAS SUBPROGRAM ELEMENT COMOBR

```
      310 DO 330 N=1,NBKOUT
          CALL BLKIN(NWDBLK,MTEMP,N,LSFILE,ISTAT)
          DO 320 M=1,23
      320 INOBS(M,N)=MTEMP(M)
      330 CONTINUE
          RETURN
          END
```

@HDG,P CFAS SUBPROGRAM ELEMENT DEPCLD

@PRT,S CFAS.DEPCLD
FURPUR 0026-10/28-13:57

CFAS SUBPROGRAM ELEMENT DEPCLD

CLOUD-FOG*CFAS.DEPCLD

```
      SUBROUTINE DEPCLD(PRES,TEMP,DEP,NCLD)
C
C     ROUTINE TO CONVERT DEWPOINT DEPRESSION, TEMPERATURE, AND
C               PRESURE INFORMATION INTO PERCENT CLOUD COVER.
C
C     CPCLD1 = CPS TO CLOUD CONVERSION TABLE AT 850 MB.
C     CPCLD2 = CPS TO CLOUD CONVERSION TABLE AT 700 MB.
C     CPCLD3 = CPS TO CLOUD CONVERSION TABLE AT 500 MB.
C     CPCLD4 = CPS TO CLOUD CONVERSION TABLE AT 300 MB.
C     PRESTD = STANDARD PRESSURE LEVELS FOR CPS TO CLOUD CONVERSION.
C     NCLD = PERCENT CLOUD COVER
C     DPRCPS = CONVERSION FACTORS FOR DEWPOINT DEPRESSION
C     TCOR = TEMPERATURE CORRECTION FOR CPS
C     PRES  = MIDPOINT PRESSURE OF CFDB LAYER, MILLIBARS
C     TEMP = MIDPOINT TEMPERATURE OF CFDB LAYER, DEG. K
C     DEP = MIDPOINT DEWPOINT DEPRESSION OF CFDB LAYER, DEG C.
C     A,B,C = CONSTANTS IN THE EXPRESSION
C               DPRCPS = A+ B*(PRESSURE/1000) + C*(PRESSURE/1000)**2
C       THIS EXPRESSION CONVERTS DEWPOINT DEPRESSION TO CONDENSATION
C       PRESSURE SPREAD CONVERSION FACTORS FOR CFDB LAYERS
C     CPS = CONDENSATION PRESSURE SPREAD OF CFDB LAYERS
C
      DIMENSION  PRES(9),TEMP(9),DEP(9),PRESTD(4),TCOR(12),CPCLD1(75),
     *CPCLD2(75),CPCLD3(75),CPCLD4(75),NCLD(9),CPS(9)

      DATA A/-4.90162240/,B/-0.93104502C/,C/-9.02129190/
      DATA MISS/-32768/
      DATA TCOR/1.05,1.10,1.15,1.20,1.25,1.30,1.37,1.5,1.75,2.0,2.4,2.8/
      DATA PRESTD/850.,700.,500.,300./
      DATA CPCLD1/
     *1.00000000, .98700000, .97600000, .96399999, .95000000,
     * .93300000, .91500000, .89500000, .87800000, .85500000,
     * .83300000, .80300000, .76400000, .72399999, .67500000,
     * .63500000, .58000000, .52500000, .52000000, .49000000,
     * .45000000, .42500000, .40000000, .37500000, .35000000,
     * .33700000, .31500000, .30000000, .27500000, .26500000,
     * .25000000, .23700000, .22200000, .20800000, .19500000,
     * .18000000, .17200000, .16200000, .15300000, .14500000,
     * .14000000, .13000000, .12500000, .11600000, .10700000,
     * .09800000, .09400000, .08200000, .07500000, .06600000,
     * .05800000, .05100000, .04400000, .03600000, .02700000,
     * .02000000, .01200000, .00800000, .00500000, .00300000,
     * .00100000,14*0./
      DATA CPCLD2/
     *1.00000000, .99699999, .98899999, .98100000, .97300000,
     * .96200000, .95200000, .93600000, .92299999, .90499999,
     * .89399999, .87500000, .84999999, .81500000, .77999999,
     * .74500000, .70500000, .66500000, .61600000, .55800000,
     * .50000000, .47800000, .45300000, .43100000, .40800000,
     * .38700000, .36900000, .35100000, .33400000, .31700000,
     * .30000000, .28600000, .27200000, .25900000, .24600000,
     * .23300000, .22200000, .21100000, .20000000, .19000000,
     * .18000000, .17100000, .16000000, .14900000, .13500000,
     * .11000000, .09400000, .07500000, .05800000, .05100000,
     * .03900000, .02900000, .02000000, .01000000,21*0./
      DATA CPCLD3/
     *1.00000000, .99500000, .98999999, .98199999, .97200000,
     * .96200000, .95200000, .94199999, .92800000, .91500000,
```

```
          * .89799999, .88000000, .86099999, .84199999, .94199999,
          * .79700000, .76700000, .72099999, .67500000, .62500000,
          * .57500000, .54600000, .51700000, .48800000, .45900000,
          * .43000000, .41900000, .40800000, .39700000, .38600000,
          * .37500000, .35200000, .32900000, .30600000, .28300000,
          * .26000000, .24800000, .23600000, .22400000, .21200000,
          * .20000000, .19000000, .18000000, .17000000, .15000000,
          * .12500000, .09800000, .07500000, .05100000, .02500000,
          *25*0./
           DATA CPCLD4/
          *1.00000000, .99500000, .98999999, .98199999, .97200000,
          * .96203049, .95200000, .94199999, .92900000, .91500000,
          * .89799999, .88000000, .86099999, .84199999, .81699999,
          * .79700000, .76700000, .72099999, .67500000, .62500000,
          * .57500000, .54600000, .51700000, .48800000, .45900000,
          * .43000000, .41900000, .40800000, .39700000, .38600000,
          * .37500000, .35200000, .32900000, .30600000, .28300000,
          * .26000000, .24800000, .23600000, .22400000, .21200000,
          * .20000000, .19000000, .15000000, .12500000, .09800000,
          * .07500000, .05100000, .02500000,27*0./
C
C          LOOP TO STEP THROUGH CFDB LAYERS.
C
           DO 200 LAY=1,9
C
C          JUMP IF TEMPERATURE NOT MISSING AND DEWPOINT DEPRESSION GE 0.
C
           IF(TEMP(LAY) .GT. 0.0 .AND. DEP(LAY) .GE. 0.0) GO TO 10
C
C          CODE LAYER AS UNKNOWN CLOUD COVER.
C
           NCLD(LAY)=MISS
C
C          CODE LAYER AS UNKNOWN CPS.
C
           CPS(LAY)=MISS
           GO TO 200
C
C          DETERMINE DEWPOINT DEPRESSION TO CPS CONVERSION FACTOR.
C
        10 DPRCPS=A+B*(PRES(LAY)/1000.)+C*(PRES(LAY)/1000.)**2
C
C          CALCULATE UNCORRECTED CPS.
C
           CPS(LAY)=DPRCPS*DEP(LAY)
C
C          DETERMINE APPROPRIATE TEMPERATURE CORRECTION FACTOR TO CPS.
C
           IF(TEMP(LAY) .GT. 268.2) GO TO 40
           IF(TEMP(LAY) .GT. 213.2) GO TO 20
           KK=12
           GO TO 30
        20 KK=-0.2*(TEMP(LAY)-273.2)
C
C          CORRECT CPS FOR TEMPERATURE
C
        30 CPS(LAY)=TCOR(KK)*CPS(LAY)
C
C          DETERMINE APPROPRIATE ENTRY IN CPS TO CLOUD TABLE.
```

CFAS SUBPROGRAM ELEMENT DEPCLD

```
C
   40 INDEX=-CPS(LAY)*0.5+1.5
C
C     IF INDEX OF CPS TO CLOUD TOO LARGE, CODE NO CLOUD
C
      IF(INDEX .LT. 75) GO TO 50
      NCLD(LAY)=0
      GO TO 200
C
C     JUMP IF PRESSURE LEVEL BELOW LOWEST LEVEL OF TABLE
C
   50 IF(PRES(LAY) .GE. PRESTD(1)) GO TO 170
C
C     JUMP IF PRESSURE LEVEL ABOVE HIGHEST LEVEL OF TABLE.
C
      IF(PRES(LAY) .LE. PRESTD(4)) GO TO 180
C
C     LOOP TO DETERMINE UPPER BOUND OF PRESSURE LEVEL.
C
      DO 60 IL=2,4
      LEV=6-IL
      IF(PRES(LAY) .LE. PRESTD(LEV)) GO TO 60
      LEVHI=LEV
   60 CONTINUE
      LEVLOW=LEVHI-1
C
C     DETERMINE CLOUD COVER OF UPPER STANDARD PRESSURE LEVEL.
C
      GO TO (70,80,90,100),LEVHI
   70 CLDHI=CPCLD1(INDEX)*100.
      GO TO 110
   80 CLDHI=CPCLD2(INDEX)*100.
      GO TO 110
   90 CLDHI=CPCLD3(INDEX)*100.
      GO TO 110
  100 CLDHI=CPCLD4(INDEX)*100.
C
C     DETERMINE CLOUD COVER OF LOWER STANDARD PRESSURE LEVEL.
C
  110 GO TO (120,130,140,150),LEVLOW
  120 CLDLOW=CPCLD1(INDEX)*100.
      GO TO 160
  130 CLDLOW=CPCLD2(INDEX)*100.
      GO TO 160
  140 CLDLOW=CPCLD3(INDEX)*100.
      GO TO 160
  150 CLDLOW=CPCLD4(INDEX)*100.
C
C     CALCULATE CLOUD COVER OF INTERMEDIATE PRESSURE LEVEL.
C
  160 NCLD(LAY)=CLDLOW+(CLDHI-CLDLOW)*(PRES(LAY)-PRESTD(LEVLOW))/
     *(PRESTD(LEVHI)-PRESTD(LEVLOW))+0.5
      GO TO 190
C
C     DETERMINE CLOUD COVER OF PRESSURE LEVEL BELOW ALL STANDARD
C        PRESSURE LEVELS.
C
  170 NCLD(LAY)=CPCLD1(INDEX)*100.+0.5
      GO TO 190
```

CFAS SUBPROGRAM ELEMENT DEPCLD

```
      C
      C     DETERMINE CLOUD COVER OF PRESSURE LEVEL ABOVE ALL STANDARD
      C         PRESSURE LEVELS.
      C
        180 NCLD(LAY)=CPCLD4(INDEX)*100.+0.5
      C
      C     ROUND CLOUD COVER TO NEAREST 5 PERCENT
      C
        190 NCLD(LAY)=NCLD(LAY)+2-MOD(NCLD(LAY)+2,5)
      C
      C     GAURD AGAINST MINUS ZERO.
      C
            NCLD(LAY)=IABS(NCLD(LAY))
        200 CONTINUE
            RETURN
            END
```

@HDG,P CFAS SUBPROGRAM ELEMENT FIND1B

@PRT,S CFAS.FIND1B
FURPUR 0026-10/28-13:57

I-32

CFAS SUBPROGRAM ELEMENT FIND1B

CLOUD-FOG*CFAS.FIND1B

```
      SUBROUTINE FIND1B (INCODE, IX, IY, RADIUS, ITMIN, ITMAX,
     * IREC, NOMORE)
C
C FIND1B IS USED WHEN THE USER WISHES TO EXAMINE ALL THE OBS/REP'S
C STORED THAT ARE WITHIN A SPECIFIED RADIUS OF SPECIFIED COORDINATES
C WHICH WERE OBSERVED DURING A SPECIFIED TIME INTERVAL.  EACH CALL TO
C FIND1B RETURNS ONE OBS/REP GOING BACKWARD IN TIME SEQUENCE.
C
C INCODE = USER CONTROL CODE.  INCODE = 1 INITIATES THE SEQUENCE AND
C          SEARCHES FOR THE NEWEST OBS/REP WHICH SATISFIES THE LOCATION
C          AND TIME REQUIREMENTS.  THIS OBS/REP IS RETURNED TO THE USER
C          IN USER BUFFER IREC.  INCODE NOT = 1 IS USED ON SUCCESSIVE
C          CALLS TO RETRIEVE THE NEXT OBS/REP IN BACKWARD TIME SEQUENCE.
C IX     = RELATIVE X POSITION IN HECTOMETERS.
C IY     = RELATIVE Y POSITION IN HECTOMETERS.
C RADIUS = RADIUS IN HECTOMETERS OF CIRCLE TO BE CENTERED AT (IX, IY).
C          ALL OBS/REP'S RETURNED TO USER WILL BE IN THIS CIRCLE.
C ITMIN  = MINIMUM, OR OLDEST, OBSERVATION TIME IN MINUTES (0-1439).
C ITMAX  = MAXIMUM, OR NEWEST, OBSERVATION TIME IN MINUTES (0-1439).
C          FIND1B WILL RETURN OBS/REP'S STARTING AT ITMAX, OR OLDER.
C IREC   = BUFFER IN CALLING ROUTINE CONTAINING NWDREC WORDS WHERE THE
C          OBS/REP WILL BE STORED.
C NOMORE = STATUS RETURNED TO USER.  NOMORE = 0 INDICATES THAT AN
C          OBS/REP WAS RETURNED TO THE USER IN IREC AND THAT THERE MAY
C          BE MORE OBS/REP'S IF THE USER SHOULD CALL AGAIN.  NOMORE = 1
C          INDICATES THAT NO OBS/REP WAS RETURNED AND THAT NO ADDITIONAL
C          OBS/REP'S EXIST IN THE DATA BASE WITHIN THE SPECIFIED TIME
C          AND LOCATION CONSTRAINTS.  THE USER SHOULD ASSUME THAT THE
C          CONTENTS OF IREC WILL BE MODIFIED WHENEVER FIND1B IS CALLED.
      COMMON /BASE/ DXSECT, DYSECT, EDGE, IBLOCK, IDTIME, IDXUTM,
     * IDYUTM, INUMBR, ISTATI, ISTATO, JNUMBR, JSTATI, JSTATO, JTIME,
     * LASTJ, MAXGPS, NBJNOW, NBLKFJ, NCOLS, NGX, NGY, NINI, NINTAB,
     * NROWS, NRPBFI, NRPBFJ, NSECTR, NWDBKI, NWDBKJ, NWDREC, NXSECT,
     * NYSECT, UTMPGD, XBASE, XMAX, XMIN, YBASE, YMAX, YMIN,
     * NNEWRS(100), NALLRS(100), ITABLE(4, 500), IBUF(3750), JBUF(1000),
     * JTIMES(100), IRXMAX, IRXMIN, IRYMAX, IRYMIN
C
      DIMENSION IREC(1)
      RADSQ=RADIUS*RADIUS
   10 CALL GET1BW (INCODE, ITMAX, IREC, NOMORE)
   20 IF (NOMORE .EQ. 1) RETURN
      IF (ITMDIF (IREC(IDTIME), ITMIN) .GE. 0) GO TO 30
      NOMORE=1
      RETURN
   30 XDIF=IX-IREC(IDXUTM)
      YDIF=IY-IREC(IDYUTM)
      IF (XDIF*XDIF+YDIF*YDIF .LE. RADSQ) RETURN
      CALL GET1BW (2, ITMAX, IREC, NOMORE)
      GO TO 20
      END
```

@HDG,P CFAS SUBPROGRAM ELEMENT FOG

@PRT,S CFAS.FOG
FURPUR 0026-10/28-13:57

CFAS SUBPROGRAM ELEMENT FOG

CLOUD-FOG*CFAS.FOG

```
      SUBROUTINE FOG(NVIS,NWEA,AMT,VALU)
C
C     ROUTINE TO CHECK FOR FOG AND MAKE DECISIONS AS TO PERCENTAGE CLOUD
C     COVER AND TOPS OF CLOUDS BASED ON HORIZONTAL VISIBILITY AND TYPE
C     OF FOG.
C
C     NVIS = HORIZONTAL VISIBILITY IN METERS
C     NWEA = SURFACE WEATHER WMO CODE 4677
C
C DERIVED LAYERED CLOUD INFORMATION
C
C  NUMLAY = NUMBER OF LAYERS GENERATED
C  KIND = KIND OF CLOUD LAYER
C              1 = LOW
C              2 = MIDDLE
C              3 = HIGH
C              4 = FOG
C              5 = LOWEST CLOUD
C              6 = CLEAR LAYER
C  ITHIN = THIN LAYER DESIGNATOR
C              MISSING = NOT THIN
C              1 = THIN
C  COVER = CLOUD COVER IN LAYER (0.0 - 1.0)
C  BASE = HEIGHT OF THE BASE OF LAYER, FEET.
C  TOP = HEIGHT OF TOP OF CLOUD LAYER, FEET.
C
C
      DIMENSION NWEA(7)

      COMMON/CLOUDS/NUMLAY,KIND(10),ITHIN(10),COVER(10),BASE(10),TOP(10)

C     SET INDICATOR FOR NO FOG AND INITIALIZE AMOUNT.
C
      IF(NUMLAY .EQ. 0) VALU=10.
      VALU=(VALU+10.)/2.
      NFOG=0
      AMT=0.
C
C     RETURN IF VISIBILITY GT 1 MILE
C
      IF(NVIS.GT.1600)RETURN
C
C     LOOP TO STEP THROUGH WEATHER
C
      DO 10 NCHK=1,7
C
C     JUMP IF NOT FOG

      IF(NWEA(NCHK).LT.40.OR.NWEA(NCHK).GT.49) GO TO 10
C
C     SET INDICATOR FOR FOG
C
      NFOG=1
C
C     DETERMINE FOG TYPE INDICATOR
C
      NTYPE=MOD(NWEA(NCHK),10)+1
      GO TO(1,1,2,5,3,5,4,5,3,5),NTYPE
```

CFAS SUBPROGRAM ELEMENT FOG

```
C
C        DETERMINE CLOUD COVER
C
       1 AMOUNT=0.125
         GO TO 9
       2 AMOUNT=0.25
         GO TO 9
       3 AMOUNT=0.5
         GO TO 9
       4 AMOUNT=0.75
         GO TO 9
       5 AMOUNT=1.0
C
C        DETERMINE MAXIMUM OF OLD AND NEW CLOUD COVER.
C
       9 AMT=AMAX1(AMT,AMOUNT)
      10 CONTINUE
C
C        RETURN IF NO FOG.
C
         IF(NFOG.EQ.0) RETURN
C
C        INCREASE NUMBER OF LAYER COUNTER, SET CLOUD COVER, AND SET BASE
C             TO ZERO.
C
         NUMLAY=NUMLAY+1
         KIND(NUMLAY)=4
         COVER(NUMLAY)=AMT
         BASE(NUMLAY)=0.
C
C        JUMP IF HORIZONTAL VISIBILITY GE 1/2 MILE OR UNKNOWN.
C
         IF(NVIS .GE. 800 .OR. NVIS .LT. 0) GO TO 11
C
C        SET TOP TO 249 FEET
C
         TOP(NUMLAY)=249.
         RETURN
C
C        SET TOP TO 149 FEET
C
      11 TOP(NUMLAY)=149.
C
C        REDUCE VALU OF FOG RELATED INFORMATION BY 3 IF VISIBILITY IS UN-
C        KNOWN.
C
         IF(NVIS .LT. 0) VALU=(((VALU*2.)-10.)+7.)/2.
         RETURN
         END
```

@HDG,P CFAS SUBPROGRAM ELEMENT GETOB1

@PRT,S CFAS.GETOB1
FURPUR 0026-10/28-13:57

CFAS SUBPROGRAM ELEMENT GETOB1

CLOUD-FOG*CFAS.GETOB1

```
      SUBROUTINE GETOBI (ITABID, IREC)
C GET AN OBS/REP FROM FILE I.
C ITABID = COLUMN INDEX OF ITABLE POINTING TO DESIRED OBS/REP.
C IREC   = BUFFER IN USER PROGRAM WHERE OBS/REP WILL BE STORED.
      COMMON /BASE/ DXSECT, DYSECT, EDGE, IBLOCK, IDTIME, IDXUTM,
     * IDYUTM, INUMBR, ISTATI, ISTATO, JNUMBR, JSTATI, JSTATO, JTIME,
     * LASTJ, MAXOPS, NBJNOW, NBLKFJ, NCOLS, NGX, NGY, NINI, NINTAB,
     * NROWS, NRPBFI, NRPBFJ, NSECTR, NWDBKI, NWDBKJ, NWDREC, NXSECT,
     * NYSECT, UTMPGD, XBASE, XMAX, XMIN, YBASE, YMAX, YMIN,
     * NNEWRS(100), NALLRS(100), ITABLE(4, 500), IBUF(3750), JBUF(1000),
     * JTIMES(100), IRXMAX, IRXMIN, IRYMAX, IRYMIN
      DIMENSION IREC(1)
      MYSECT=ITABLE(4, ITABID)/100
      IF (MYSECT .EQ. IBLOCK) GO TO 10
      CALL BLKIN (NWDBKI, IBUF, MYSECT, INUMBR, ISTATI)
      IBLOCK=MYSECT
   10 MYRECD=ITABLE(4, ITABID)-MYSECT*100
      INDEX=(MYRECD-1)*NWDREC
      DO 20 I=1, NWDREC
      INDEX=INDEX+1
   20 IREC(I)=IBUF(INDEX)
      RETURN
      END
```

@HDG,P CFAS SUBPROGRAM ELEMENT GET1BW

@PRT,S CFAS.GET1BW
FURPUR 0026-10/28-13:57

CFAS SUBPROGRAM ELEMENT GET1BW

CLOUD-FOG*CFAS.GET1BW

```
          SUBROUTINE GET1BW (INCODE, NTIME, IREC, NOMORE)
C GET1BW IS USED WHEN THE USER WISHES TO EXAMINE ALL THE OBS/REP'S
C STORED STARTING AT NTIME AND GOING BACKWARD IN TIME SEQUENCE.
C INCODE = USER CONTROL CODE.  INCODE = 1 INITIATES THE SEQUENCE AND
C            SEARCHES FOR THE FIRST RECORD WHICH IS RETURNED TO THE USER.
C            INCODE NOT = 1 IS USED ON SUCCESSIVE CALLS TO RETRIEVE THE
C            NEXT OBS/REP IN TIME SEQUENCE.
C NTIME  = START TIME IN MINUTES (0-1439).
C IREC   = BUFFER IN CALLING ROUTINE CONTAINING NWDREC WORDS WHERE THE
C            OBS/REP WILL BE STORED.
C NOMORE = STATUS RETURNED TO USER.  NOMORE = 0 INDICATES THAT AN
C            OBS/REP WAS RETURNED TO THE USER IN IREC AND THAT THERE MAY
C            BE MORE OBS/REP'S IF THE USER SHOULD CALL AGAIN.  NOMORE = 1
C            INDICATES THAT NO OBS/REP WAS RETURNED AND THAT NO ADDITIONAL
C            OBS/REP'S EXIST IN THE DATA BASE.
      COMMON /BASE/ DXSECT, DYSECT, EDGE, IBLOCK, IDTIME, IDXUTM,
     * IDYUTM, INUMBR, ISTATI, ISTATO, JNUMBR, JSTATI, JSTATO, JTIME,
     * LASTJ, MAXGPS, NBJNOW, NBLKFJ, NCOLS, NGX, NGY, NINI, NINTAB,
     * NROWS, NRPBFI, NRPBFJ, NSECTR, NWDBKI, NWDBKJ, NWDREC, NXSECT,
     * NYSECT, UTMPGD, XBASE, XMAX, XMIN, YBASE, YMAX, YMIN,
     * NNEWRS(100), NALLRS(100), ITABLE(4, 500), IBUF(3750), JBUF(1000),
     * JTIMES(100), IRXMAX, IRXMIN, IRYMAX, IRYMIN
      DIMENSION IREC(1)
      NOMORE=0
      IF (INCODE .NE. 1) GO TO 100
      IF (NINI .GT. 0) GO TO 40
   10 IF (INCODE .NE. 1) GO TO 30
      PRINT 20, NTIME
   20 FORMAT (1H , ' GET1BW - NO DATA RECORDS EXIST FOR TIMES LESS THAN,
     * OR EQUAL TO', I5, ' MINUTES')
   30 NOMORE=1
      RETURN
   40 IF (ITMDIF (NTIME, ITABLE(1, NINI)) .GE. 0) GO TO 80
   50 IF (NBJNOW .EQ. 0) GO TO 10
      INI=0
      JBKEND=LASTJ-NBJNOW+1
      IF (JBKEND .LT. 1) JBKEND=NBJNOW+JBKEND
      JBKGET=LASTJ
   60 CALL BLKIN (NWDBKJ, JBUF, JBKGET, JNUMBR, JSTATI)
      JRCGET=NRPBFJ
   70 INDEX=(JRCGET-1)*NWDREC+IDTIME
      IF (ITMDIF (NTIME, JBUF(INDEX)) .GE. 0) GO TO 100
      JRCGET=JRCGET-1
      IF (JRCGET .GT. 0) GO TO 70
      IF (JBKGET .EQ. JBKEND) GO TO 10
      JBKGET=JBKGET-1
      IF (JBKGET .EQ. 0) JBKGET=NBLKFJ
      GO TO 60
   80 INI=1
      IGET=0
   90 IGET=IGET+1
      IF (ITMDIF (NTIME, ITABLE(1, IGET)) .LT. 0) GO TO 90
  100 IF (INI .NE. 1) GO TO 110
      IF (IGET .GT. NINI) GO TO 50
      CALL GETOBI (IGET, IREC)
      IGET=IGET+1
      RETURN
  110 IF (JRCGET .GT. 0) GO TO 120
```

CFAS SUBPROGRAM ELEMENT GET1BW

```
          IF (JBKGET .EQ. JBKEND) GO TO 10
          JBKGET=JBKGET-1
          IF (JBKGET .EQ. 0) JBKGET=NBLKFJ
          CALL BLKIN (NWDBKJ, JBUF, JBKGET, JNUMBR, JSTATI)
          JRCGET=NRPBFJ
      120 INDEX=(JRCGET-1)*NWDREC
          DO 130 I=1, NWDREC
          INDEX=INDEX+1
      130 IREC(I)=JBUF(INDEX)
          JRCGET=JRCGET-1
          RETURN
          END
```

@HDG,P CFAS SUBPROGRAM ELEMENT GET1FW

@PRT,S CFAS.GET1FW
FURPUR 0026-10/28-13:57

CFAS SUBPROGRAM ELEMENT GET1FW

CLOUD-FOG•CFAS.GET1FW

```
      SUBROUTINE GET1FW (INCODE, NTIME, IREC, NOMORE)
C GET1FW IS USED WHEN THE USER WISHES TO EXAMINE ALL THE OBS/REP'S
C STORED STARTING AT NTIME AND GOING FORWARD IN TIME SEQUENCE.
C INCODE = USER CONTROL CODE.  INCODE = 1 INITIATES THE SEQUENCE AND
C          SEARCHES FOR THE FIRST RECORD WHICH IS RETURNED TO THE USER.
C          INCODE NOT = 1 IS USED ON SUCCESSIVE CALLS TO RETRIEVE THE
C          NEXT OBS/REP IN TIME SEQUENCE.
C NTIME  = START TIME IN MINUTES (0-1439).
C IREC   = BUFFER IN CALLING ROUTINE CONTAINING NWDREC WORDS WHERE THE
C          OBS/REP WILL BE STORED.
C NOMORE = STATUS RETURNED TO USER.  NOMORE = 0 INDICATES THAT AN
C          OBS/REP WAS RETURNED TO THE USER IN IREC AND THAT THERE MAY
C          BE MORE OBS/REP'S IF THE USER SHOULD CALL AGAIN.  NOMORE = 1
C          INDICATES THAT NO OBS/REP WAS RETURNED AND THAT NO ADDITIONAL
C          OBS/REP'S EXIST IN THE DATA BASE.
      COMMON /BASE/ DXSECT, DYSECT, EDGE, IBLOCK, IDTIME, IDXUTM,
     * IDYUTM, INUMBR, ISTATI, ISTATO, JNUMBR, JSTATI, JSTATO, JTIME,
     * LASTJ, MAXGPS, NBJNOW, NBLKFJ, NCOLS, NGX, NGY, NINI, NINTAB,
     * NROWS, NRPBFI, NRPBFJ, NSECTR, NWDBKI, NWDBKJ, NWDREC, NXSECT,
     * NYSECT, UTMPGD, XBASE, XMAX, XMIN, YBASE, YMAX, YMIN,
     * NNEWRS(100), NALLRS(100), ITABLE(4, 500), IBUF(3750), JBUF(1000),
     * JTIMES(100), IRXMAX, IRXMIN, IRYMAX, IRYMIN
      DIMENSION IREC(1)
      NOMORE=0
      IF (INCODE .NE. 1) GO TO 135
      IGET=NINI
      IF (NBJNOW .EQ. 0) GO TO 60
      IF (ITMDIF (NTIME, JTIME) .GT. 0) GO TO 60
      INI=0
      JBKGET=LASTJ-NBJNOW+1
      IF (JBKGET .LT. 1) JBKGET=NBJNOW+JBKGET
   10 IF (ITMDIF (NTIME, JTIMES(JBKGET)) .LE. 0) GO TO 20
      JBKGET=JBKGET+1
      IF (JBKGET .GT. NBLKFJ) JBKGET=1
      GO TO 10
   20 CALL BLKIN (NWDBKJ, JBUF, JBKGET, JNUMBR, JSTATI)
      JRCGET=1
   30 INDEX=(JRCGET-1)*NWDREC+IDTIME
      IF (ITMDIF(NTIME, JBUF(INDEX)) .LE. 0) GO TO 40
      JRCGET=JRCGET+1
      GO TO 30
   40 INDEX=(JRCGET-1)*NWDREC
      DO 50 I=1, NWDREC
      INDEX=INDEX+1
   50 IREC(I)=JBUF(INDEX)
      JRCGET=JRCGET+1
      RETURN
   60 INI=1
   70 IF (IGET .GT. 0) GO TO 100
      PRINT 80, NTIME
   80 FORMAT (1H , ' GET1FW - NO DATA RECORDS EXIST FOR TIMES GREATER TH
     *AN, OR EQUAL TO', I5, ' MINUTES')
   90 NOMORE=1
      RETURN
  100 IF (ITMDIF (NTIME, ITABLE(1, IGET)) .LE. 0) GO TO 110
      IGET=IGET-1
      GO TO 70
  110 CALL GETOBI (IGET, IREC)
```

CFAS SUBPROGRAM ELEMENT GET1FW

```
      IGET=IGET-1
      RETURN
  135 IF (INI .NE. 0) GO TO 150
      IF (JRCGET .LE. NRPBFJ) GO TO 40
      IF (JBKGET .NE. LASTJ) GO TO 140
      INI=1
      GO TO 110
  140 JBKGET=JBKGET+1
      IF (JBKGET .GT. NBLKFJ) JBKGET=1
      GO TO 20
  150 IF (IGET .GT. 0) GO TO 110
      GO TO 90
      END
```

@HDG,P CFAS SUBPROGRAM ELEMENT ITMDIF

@PRT,S CFAS.ITMDIF
FURPUR 0026-10/28-13:57

CFAS SUBPROGRAM ELEMENT ITMDIF

```
CLOUD-FOG*CFAS.ITMDIF
           FUNCTION ITMDIF (ITA, ITB)
      C COMPUTES DIFFERENCE BETWEEN TIMES ITA AND ITB.  RESULT IS POSITIVE IF
      C ITA IS MORE RECENT THAN ITB.  IT IS ASSUMED THAT ALL TIME DIFFERENCES
      C WILL BE LESS THAN OR EQUAL TO 720 MINUTES.
           IDIF=ITA-ITB
           IF (IDIF) 20, 30, 10
        10 IF (IDIF .LE. 720) GO TO 30
           IDIF=IDIF-1440
           GO TO 30
        20 IF (IDIF+720 .GE. 0) GO TO 30
           IDIF=IDIF+1440
        30 ITMDIF=IDIF
           RETURN
           END
@HDG,P CFAS SUBPROGRAM ELEMENT ITOJ

@PRT,S CFAS.ITOJ
FURPUR 0026-10/28-13:58
```

AD-A048 564 SPERRY RESEARCH CENTER SUDBURY MASS F/G 4/2
DEVELOPMENT OF CLOUD/FOG ANALYSIS AND APPLICATION SUBROUTINES F--ETC(U)
NOV 75 B R FOW, W D MOUNT DAAD07-74-C-0251
UNCLASSIFIED SCRC-CR-75-17 NL

3 OF 3
ADA048564

END
DATE
FILMED
2-78
DDC

CFAS SUBPROGRAM ELEMENT ITOJ

CLOUD-FOG*CFAS.ITOJ

```
      SUBROUTINE ITOJ
C DELETE THE OLDEST (NRPBFJ) RECORDS FROM FILE I AND STORE THEM AS A
C BLOCK IN FILE J.
      COMMON /BASE/ DXSECT, DYSECT, EDGE, IBLOCK, IDTIME, IDXUTM,
     * IDYUTM, INUMBR, ISTATI, ISTATO, JNUMBR, JSTATI, JSTATO, JTIME,
     * LASTJ, MAXGPS, NBJNOW, NBLKFJ, NCOLS, NGX, NGY, NINI, NINTAB,
     * NROWS, NRPBFI, NRPBFJ, NSECTR, NWDBKI, NWDBKJ, NWDREC, NXSECT,
     * NYSECT, UTMPGD, XBASE, XMAX, XMIN, YBASE, YMAX, YMIN,
     * NNEWRS(100), NALLRS(100), ITABLE(4, 500), IBUF(3750), JBUF(1000),
     * JTIMES(100), IRXMAX, IRXMIN, IRYMAX, IRYMIN
      ILOW=NINI-NRPBFJ+1
      JTIME=ITABLE(1, ILOW)
      DO 50 NRTEST=ILOW, NINI
      ITEMP=ITABLE(4, NRTEST)
      IF (ITEMP .EQ. 0) GO TO 50
      MYSECT=ITEMP/100
      IF (MYSECT .EQ. IBLOCK) GO TO 10
      CALL BLKIN (NWDBKI, IBUF, MYSECT, INUMBR, ISTATI)
      IBLOCK=MYSECT
   10 IBT100=IBLOCK*100
      DO 40 NRCHCK=NRTEST, NINI
      ITEMP=ITABLE(4, NRCHCK)
      IF (ITEMP/100 .NE. IBLOCK) GO TO 40
      IWDEND=(ITEMP-IBT100)*NWDREC
      IWDSTR=IWDEND-NWDREC+1
      INDEX=(NINI-NRCHCK)*NWDREC
      DO 20 IWD=IWDSTR, IWDEND
      INDEX=INDEX+1
   20 JBUF(INDEX)=IBUF(IWD)
      ITABLE(4, NRCHCK)=0
      NNEWRS(IBLOCK)=NNEWRS(IBLOCK)-1
   40 CONTINUE
   50 CONTINUE
      IF (NBJNOW .EQ. 0) LASTJ=0
      LASTJ=LASTJ+1
      IF (LASTJ .GT. NBLKFJ) LASTJ=1
      CALL BLKOUT (NWDBKJ, JBUF, LASTJ, JNUMBR, JSTATO)
      JTIMES(LASTJ)=JTIME
      IF (NBJNOW .LT. NBLKFJ) NBJNOW=NBJNOW+1
      NINI=NINI-NRPBFJ
      RETURN
      END
```

@HDG,P CFAS SUBPROGRAM ELEMENT LAYCLD

@PRT,S CFAS.LAYCLD
FURPUR 0026-10/28-13:58

I-42

CFAS SUBPROGRAM ELEMENT LAYCLD

CLOUD-FOG*CFAS.LAYCLD

```
      SUBROUTINE LAYCLD(DLAT,VALU)
C
C     ROUTINE TO CONSTRUCT CLOUD LAYERS FROM LAYERED CLOUD DATA IN
C           AIRWAYS, METAR, AND SYNOP TYPE OBS/REP.
C
C          LIST OF ARGUMENTS
C
C  INPUT
C
C     DLAT=LATITUDE OF OBS/REP,DEGREES (NEGATIVE IF SOUTH)
C
C  OUTPUT
C
C     VALU=INFORMATION VALU OF OBS/REP
C
C              COMMON DATA
C
C  IN
C
C     NS(J)=SKY COVER DUE TO CLOUD IN LAYER, 0-9. 1 TO 10 LAYERS.
C     ICTS=TYPE OF CLOUD IN LAYER, 0-9    WMO CODE 0500
C     IHS(J)=HEIGHT OF BASE OF CLOUD LAYER
C         AIRWAYS - 100'S OF FEET
C         METAR - WMO CODE 1677
C         SYNOP - WMO CODE 1677
C     ITHIN(J)=CLOUD LAYER THICKNESS INDICATOR
C         1 IF THIN
C         MISSING IF NOT THIN
C     ITYPE=TYPE OF OBS/REP
C           1=AIRWAYS   -1 IF A SPECIAL
C           2=METAR     -2 IF A SPECI (SPECIAL)
C           3=SYNOP
C
C  OUT
C
C     NUMLAY=NUMBER OF CLOUD LAYERS IDENTIFIED
C     KIND=KIND OF CLOUD LAYER
C              1=LOW
C              2=MIDDLE
C              3=HIGH
C              4=FOG
C              5=LOWEST CLOUD
C              6=CLEAR LAYER
C     ITHIN=THIN LAYER DESIGNATOR
C              MISSING=NOT THIN
C              1=THIN
C     COVER=FRACTION OF SKY COVERED BY CLOUDS IN THE LAYER (0.0- 1.0)
C     BASE=HEIGHT OF THE BASE OF CLOUD LAYER, FEET.
C     TOP=HEIGHT OF THE TOP OF THE CLOUD LAYER, FEET.
C
      COMMON /OBSREP/ IX,IY,IZ,ITIME,IOBC,ITYPE,IVALU,NTCLC,NCEIL,NVV,
     *MINBAS,MAXTOP,MSPWE,LCOV(9),ICL,ITSC,ICM,ICH,ICTS(10),NWEA(7),IPW,
     *IDD,IFF,IPPP,ITT,ITD,IVIS,NH,IH,NS(10),IHS(10),ITHN(10),ICLG,ICLOV
     *,IVISC,NOUSE(58)

      COMMON /CLOUDS/ NL,KIND(10),ITHIN(10),COVER(10),BASE(10),TOP(10)

      DIMENSION SBAS(3)
```

CFAS SUBPROGRAM ELEMENT LAYCLD

```
      EQUIVALENCE (SBAS(1),BASLOW), (SBAS(2),BASMID), (SBAS(3),BASHI)

      DATA
     *MISS/-32768/,
     *FMISS/-32768./,
     *BLMX/6500./,
     *BMMX/15000./

C     TOPCLR=ASSUMED TOP OF ALL CLOUD LAYERS

      TOPCLR=40000.

C     ASSIGN STANDARD BASE HEIGHTS FOR LOW AND MIDDLE CLOUDS

      BASLOW=2200.
      BASMID=11700.

C     CALCULATE A STANDARD HIGH CLOUD BASE FROM LATITUDE OF THE OBS/REP

      BASHI=35000. - 13000.*(ABS(DLAT)/90.)

C     INITIALIZE PARAMETERS.

      MT=IABS(ITYPE)
      VALU=0.
      LSC=1
      NL=0
   70 IF(NS(LSC)) 80,90,100

C     RETURN OF NO LAYERED CLOUD DATA

   80 RETURN

C     CONSTRUCT A CLEAR LAYER TO TOP

   90 NL=NL+1
      KIND(NL)=6
      COVER(NL)=0.
      BASE(NL)=0.
      TOP(NL)=TOPCLR

C     CALCULATE OBS/REP VALUE

      VALU=VALU+ 10.
      XLSC=LSC
      VALU=VALU/XLSC
      RETURN

C     JUMP TO 230 IF NOT AN OBSCURING LAYER.
C     JUMP TO 120 IF AN OBSCURING LAYER.
C     JUMP TO 105 IF SKY COVER IS NOT IN RANGE 0 TO 9

  100 IF(NS(LSC)-9) 230,120,105

C     DIMINISH VALUE OF OBS/REP BECAUSE OF PROBABLE ERROR, THEN RETURN

  105 XLSC=LSC
      VALU=VALU/XLSC
```

CFAS SUBPROGRAM ELEMENT LAYCLD

```
          RETURN
C         CONSTRUCT A TOTAL OVERCAST LAYER
      120 NL=NL+1
          KIND(NL)=1
          COVER(NL)=1.
          VALU=VALU+10.
C         DIMINISH VALU IF BASE HEIGHT NOT GIVEN JUMP TO 130 IF GIVEN
          IF (IHS(LSC) .GE. 0) GO TO 130
          IHS(LSC)=-32768
          VALU=VALU-3.
          GO TO 180
      130 IF(MT .NE. 1) GO TO 140
          BASE(NL)=IHS(LSC)*100
C         SET BASE OF OVERCAST LAYER EQUAL TO THE SMALLER OF THE CALCULATED
C            VALUE OR THE ASSUMED HIGH CLOUD BASE HEIGHT
          BASE(NL)=AMIN1(BASHI,BASE(NL))
          GO TO 190
      140 IF(IHS(LSC) .GT. 50) GO TO 150
          BASE(NL)=IHS(LSC)*100
          GO TO 190
      150 IF(IHS(LSC) .GT. 80) GO TO 160
          BASE(NL)=(IHS(LSC)-50)*1000
          GO TO 190
C         JUMP TO 170 IF CLOUD LAYER BASE HEIGHT HIGHER THAN HIGH CLOUD BASE
      160 IF(IHS(LSC) .LT. 90) GO TO 170
C         CLOUD LAYER BASE HEIGHT OUT OF ALLOWABLE RANGE, PROBABLE ERROR
C         REDUCE VALU BY A TOTAL OF 5 AND USE THE STANDARD HIGH CLOUD BASE
          VALU=VALU-3.
      170 VALU=VALU-2.
          BASE(NL)=BASHI
          GO TO 190
      180 BASE(NL)=BASLOW
C         CONSTRUCT A CLEAR LAYER TO THE BASE OF THE OVERCAST LAYER
      190 NL=NL+1
          KIND(NL)=6
          COVER(NL)=0.
          TOP(NL)=BASE(NL-1)
          BASE(NL)=0.
          XLSC=LSC
          VALU=VALU/XLSC
          RETURN
C         COME HERE IF NOT AN OBSCURING LAYER.
      230 NL=NL+1
          VALU=VALU+10.
          COVER(NL)=FLOAT(NS(LSC))/8.
```

CFAS SUBPROGRAM ELEMENT LAYCLD

```
      C      JUMP TO 330 IF CLOUD LAYER HEIGHT IS NOT MISSING

             IF(IHS(LSC) .GT. 0) GO TO 330

      C      JUMP TO 280 FOR SYNOP AND METAR CODES

             IF(MT .NE. 1) GO TO 280

      C      JUMP TO 270 IF SKY COVER OF NEXT LAYER IF ANY IS NOT MISSING

         240 IF(NS(LSC+1) .GE. 0 .AND. NS(LSC+1) .LE. 9) GO TO 270

      C      JUMP TO 250 IF OTHER LAYERS HAVE BEEN CONSTRUCTED.

             IF(NL .GT. 1) GO TO 250
             VALU=0.
             NL=0
             COVER(NL)=FMISS
             RETURN

      C      DETERMINE THE KIND OF HIGHEST CLOUD LAYER YET CONSTRUCTED

         250 HBASE=BASE(1)
             KMX=KIND(1)
             DO 252 IJ=1,NL
             IF(BASE(NL) .LT. HBASE) GO TO 252
             KMX=KIND(IJ)
             HBASE=BASE(IJ)
         252 CONTINUE

      C      JUMP TO 260 IF  KIND  OF HIGHEST CLOUD LAYER IS 1 OR 2

             IF(KMX .LE. 2) GO TO 260

      C      HIGHEST LAYER CONSTRUCTED THUS FAR WAS A HIGH TYPE CLOUD, PROBABLE
      C           ERROR IN DATA, DISREGARD PRESENT LAYER AND REDUCE VALU

             NL=NL-1
             VALU=VALU-5.
             GO TO 460
         260 KIND(NL)=KMX+1
             BASE(NL)=SBAS(KMX+1)
             IF(ITHN(LSC) .EQ. 1) ITHIN(NL)=1
             VALU=VALU-2.5
             GO TO 460

      C      DISREGARD DATA ON PRESENT LAYER

         270 NL=NL-1
             VALU=VALU-10.
             GO TO 450

      C      METAR AND SYNOP OBS/REP WITH MISSING BASE HEIGHTS COME HERE. JUMP
      C           TO 240 IF CLOUD TYPE NOT GIVEN

         280 IF(ICTS(LSC) .LT. 0 .OR. ICTS(LSC) .GT. 9) GO TO 240

      C      DETERMINE BASE OF LAYER FROM CLOUD TYPE
```

CFAS SUBPROGRAM ELEMENT LAYCLD

```
      290 IF(ICTS(LSC) .GT. 2) GO TO 310
          KIND(NL)=3
          BASE(NL)=BASHI
          VALU=VALU-2.
          GO TO 450
      310 IF(ICTS(LSC) .GT. 5) GO TO 320
          KIND(NL)=2
          BASE(NL)=BASMID
          VALU=VALU-2.
          GO TO 450
      320 KIND(NL)=1
          BASE(NL)=BASLOW
          VALU=VALU-2.
          GO TO 450
C         COME HERE IF BASE HEIGHT CODE IS NOT MISSING
      330 IF(IHS(LSC) .GT. 50) GO TO 340
          BASE(NL)=IHS(LSC)*100
C         AIRWAYS OBS/REP JUMP TO 420 AND DETERMINE  KIND  FROM BASE HEIGHT
          IF(MT .NE. 1) GO TO 390
C         METAR AND SYNOP OBS/REP JUMP TO 390 TO DETERMINE  KIND  FROM CLOUD
          IF(ITHN(LSC) .EQ. 1) ITHIN(NL)=1
          GO TO 420
      340 IF(IHS(LSC) .GT. 80) GO TO 360
          IF(MT .NE. 1) GO TO 350
          BASE(NL)=IHS(LSC)*100
          IF(ITHN(LSC) .EQ. 1) ITHIN(NL)=1
          GO TO 420
      350 BASE(NL)=(IHS(LSC)-50)*1000
          GO TO 390
      360 IF(MT .NE. 1) GO TO 380
          BASE(NL)=IHS(LSC)*100
          IF(BASE(NL) .LE. 30000.) GO TO 370
C         PROBABLE ERROR IN OBS/REP. USE ASSUMED HIGH BASE AND REDUCE VALU
          BASE(NL)=BASHI
          VALU=VALU-2.
      370 IF(ITHN(LSC) .EQ. 1) ITHIN(NL)=1
          GO TO 420
C         PROBABLE ERROR IN OBS/REP. USE ASSUMED HIGH BASE AND REDUCE VALU
      380 BASE(NL)=BASHI
          VALU=VALU-2.
C         JUMP TO 420 IF CLOUD TYPE MISSING OR NOT IN ALLOWABLE RANGE
      390 IF(ICTS(LSC) .LT. 0 .OR. ICTS(LSC) .GT. 9) GO TO 420
          IF(ICTS(LSC) .GT. 2) GO TO 400
          KIND(NL)=3
          GO TO 420
      400 IF(ICTS(LSC) .GT. 5) GO TO 410
```

CFAS SUBPROGRAM ELEMENT LAYCLD

```
            KIND(NL)=2
            GO TO 420
      410   KIND(NL)=1
            GO TO 420

C           AIRWAYS AND SYNOP OR METAR OBS/REP WITH MISSING CLOUD TYPES COME
C                 HERE TO DETERMINE LAYER KIND. ALSO COME HERE TO CHECK LAYER
C                 KIND AS DETERMINED FROM CLOUD TYPE. LAYER KIND AS DETERMINED
C                 FROM BASE HEIGHT OVERRIDES DETERMINATION FROM CLOUD TYPE. RE-
C                 DUCE VALU BY 2. IF THE TWO DETERMINATIONS OF KIND DO NOT
C                 AGREE.

      420   IF(BASE(NL) .GT. BLMX) GO TO 430
            IF(KIND(NL) .EQ. -32768) GO TO 425
            IF(KIND(NL) .NE. 1) VALU=VALU-2.
      425   KIND(NL)=1
            GO TO 450
      430   IF(BASE(NL) .GT. BMMX) GO TO 440
            IF(KIND(NL) .EQ. -32768) GO TO 435
            IF(KIND(NL) .NE. 2) VALU=VALU-2.
      435.  KIND(NL)=2
            GO TO 450
      440   IF(KIND(NL) .EQ. -32768) GO TO 445
            IF(KIND(NL) .NE. 3) VALU=VALU-2.
      445   KIND(NL)=3

C           TEST FOR OVERCAST PRESENT LAYER, IF NOT, TEST FOR MORE LAYERED
C                 CLOUD DATA.

      450   IF(NS(LSC) .GE. 8) GO TO 470
            LSC=LSC+1
            IF(NS(LSC)) 455,465,490
      455   LSC=LSC-1
      460   IF(NS(LSC) .GE. 8) GO TO 470

C           CONSTRUCT A CLEAR LAYER FROM SURFACE TO TOP WHEN LAST LAYER NOT
C           TOTALLY OVERCAST.

      465   NL=NL+1
            KIND(NL)=6
            COVER(NL)=0.
            BASE(NL)=0.
            TOP(NL)=TOPCLR
            GO TO 480

C           CONSTRUCT A CLEAR LAYER TO THE BASE OF OVERCAST OR OBSCURING LAYER
C                  WHEN EITHER OF THESE WAS THE LAST LAYER.

      470   NL=NL+1
            KIND(NL)=6
            COVER(NL)=0.
            BASE(NL)=0.
            TOP(NL)=BASE(NL-1)
      480   XLSC=LSC
            VALU=VALU/XLSC
            RETURN
      490   IF(NS(LSC)-9) 230,500,455
      500   NS(LSC)=8
            GO TO 230
            END
```

```
CFAS SUBPROGRAM ELEMENT MVLCOV

CLOUD-FOG*CFAS.MVLCOV
      SUBROUTINE MVLCOV (LCOVA, LCOVB, IHA, IHB)
C
C     THIS ROUTINE CALCULATES THE CLOUD COVER IN THE CFDB LAYERS OF A
C     STATION 'A', LCOVA(I), AT AN ELEVATION OF IHA (METERS) THAT WOULD
C     EXIST IF THE LAYERED CLOUD COVERAGE AT A STATION 'B', LCOVB(I), OF
C     ELEVATION IHB (METERS) WERE MOVED TO 'A' WITH THE CFDB LAYERS OF
C     'B' RETAINING THEIR REFERENCE LEVEL, IHB.
C
C     INPUT DATA
C
C     LCOVB(I) = CLOUD COVER IN THE CFDB LAYERS OF STATION 'B'
C     IHB = HEIGHT ABOVE MEAN SEA LEVEL OF STATION 'B'
C     IHA = HEIGHT ABOVE MEAN SEA LEVEL OF STATION 'A'
C
C     OUTPUT DATA
C
C     LCOVA(I) = CLOUD COVER IN THE CFDB LAYERS OF STATION 'A'
C
      DIMENSION LCOVA(9), LCOVB(9), LEVELS(10)
      DATA LEVELS/0, 150, 300, 600, 1000, 2000, 3500, 5000, 6500, 10000/
      DATA MISS /-32768/
      IHDIF=(IHA-IHB)*3.281+0.5
      DO 40 LEVELA=1, 9
      MINA=LEVELS(LEVELA)+IHDIF
      MAXA=LEVELS(LEVELA+1)+IHDIF
      NPARTS=0
      ISUM1=0
      SUMFT=0.0
      SUM5XF=0.0
      DO 10 LEVELB=1, 9
      IF (LEVELS(LEVELB) .GT. MAXA) GO TO 20
      IF (LEVELS(LEVELB+1) .LT. MINA) GO TO 10
      IF (LCOVB(LEVELB) .EQ. MISS) GO TO 10
      MINAB=MINA
      IF (LEVELS(LEVELB) .GT. MINAB) MINAB=LEVELS(LEVELB)
      MAXAB=MAXA
      IF (LEVELS(LEVELB+1) .LT. MAXAB) MAXAB=LEVELS(LEVELB+1)
      IFTDIF=MAXAB-MINAB
      IF (IFTDIF .LE. 0) GO TO 10
      NPARTS=NPARTS+1
      MULT5=LCOVB(LEVELB)/5
      ISUM1=ISUM1+LCOVB(LEVELB)-MULT5*5
      SUMFT=SUMFT+IFTDIF
      SUM5XF=SUM5XF+MULT5*IFTDIF
   10 CONTINUE
   20 IF (NPARTS .NE. 0) GO TO 30
      LCOVA(LEVELA)=MISS
      GO TO 40
   30 LCOV=(SUM5XF/SUMFT)*5.0+2.5
      LCOV=(LCOV/5)*5
      SUM1=ISUM1
      IEXTRA=SUM1/NPARTS+0.5
      LCOVA(LEVELA)=LCOV+IEXTRA
   40 CONTINUE
      RETURN
      END

@HDG.P CFAS SUBPROGRAM ELEMENT NOSECT
```

CFAS SUBPROGRAM ELEMENT NOSECT

```
@PRT,S CFAS.NOSECT
FURPUR 0026-10/28-13:58
```

CFAS SUBPROGRAM ELEMENT NOSECT

CLOUD-FOG*CFAS.NOSECT

```
            FUNCTION NOSECT (IX, IY)
      C COMPUTES SECTOR NO (1-NSECTR) FROM UTM COORDINATES (IY, IX).
            COMMON /BASE/ DXSECT, DYSECT, EDGE, IBLOCK, IDTIME, IDXUTM,
           * IDYUTM, INUMBR, ISTATI, ISTATO, JNUMBR, JSTATI, JSTATO, JTIME,
           * LASTJ, MAXGPS, NBJNOW, NBLKFJ, NCOLS, NGX, NGY, NINI, NINTAB,
           * NROWS, NRPBFI, NRPBFJ, NSECTR, NWDBKI, NWDBKJ, NWDREC, NXSECT,
           * NYSECT, UTMPGD, XBASE, XMAX, XMIN, YBASE, YMAX, YMIN,
           * NNEWRS(100), NALLRS(100), ITABLE(4, 500), IBUF(3750), JBUF(1000),
           * JTIMES(100), IRXMAX, IRXMIN, IRYMAX, IRYMIN
            MYROW=(IY+YBASE-YMIN)/DYSECT
            MYCOL=(IX+XBASE-XMIN)/DXSECT
            NOSECT=MYCOL*NYSECT+MYROW+1
            RETURN
            END
```

@HDG,P CFAS SUBPROGRAM ELEMENT RAOB

@PRT,S CFAS.RAOB
FURPUR 0026-10/28-13:58

CFAS SUBPROGRAM ELEMENT RAOB

CLOUD-FOG*CFAS.RAOB

```
      SUBROUTINE RAOB(HMP,PMP,TMP,DMP,VALU)
C
C     ROUTINE TO CALCULATE TEMPERATURE, DEWPOINT DEPRESSION, AND PRES-
C.    SURE FOR THE MIDPOINT OF THE CFDB LAYERS
C
C     INPUT DATA
C     IX     = X DISTANCE OF RAOB SITE FROM IXREF, HECTOMETERS.
C     IY     = Y DISTANCE OF RAOB SITE FROM IYREF, HECTOMETERS.
C     IH     = TERRAIN HEIGHT AT RAOB SITE, METERS
C     ITIME  = TIME OF RAOB (0-1439)
C     ITYPE  = 4, (-4 IF A SPECIAL RAOB)
C     IZ(I)  = ALTITUDE OF RAOB REPORTING LEVEL, DEKAMETERS
C     IP(I)  = PRESSURE OF RAOB REPORTING LEVELS, MILLIBARS*10
C     IT(I)  = TEMPERATURE OF RAOB REPORTING LEVEL, (DEG. K.)*10
C     IDD(I) = DEWPOINT DEPRESSION OF RAOB REPORTING LEVEL, (DEG. C)*10
C     NRRL   = NUMBER OF RAOB REPORTING LEVELS
C     HMP(J) = HEIGHT ABOVE MEAN SEA LEVEL OF MIDPOINT OF CFDB LAYERS,
C              METERS.
C     PMP(J) = PRESSURE AT MIDPOINT OF THE CFDB LAYERS, MILLIBARS.
C     TMP(J) = TEMPERATURE AT MIDPOINT OF THE CFDB LAYERS, DEG. K.
C     DMP(J) = DEWPOINT DEPRESSION AT MIDPOINT OF THE CFDB LAYER, DEG. K.
C
C               **** THIS ROUTINE ASSUMES ****
C     1. PRESSURES ARE IN DECREASING ORDER
C     2. STATION ELEVATION IS GIVEN
C     3. TEMPERATURE AT TOP RAOB LEVEL IS GIVEN
C     4. TEMPERATURE AT TWO RAOB LEVELS ARE GIVEN
C     5. FIRST RAOB LEVEL IS AT SURFACE
C     6. ALL PRESSURES (EXCEPT SURFACE) ARE GIVEN
C     7. MISSING DATA WORDS ARE FILLED WITH -32768
C     DEFINITIONS OF FREQUENTLY USED VARIABLE NAMES
C     LEVHGT = RAOB LEVEL NO. OF LOWEST HEIGHT
C
      COMMON /OBSREP/ IX,IY,IH,ITIME,IOBC,ITYPE,IVALU,NU(3),MINBAS,
     *MAXTOP,NLV,LCOV(9),IZ(30),IP(30),IT(30),IDD(30),NRRL
      DIMENSION HMP(9),PMP(9),TMP(9),DMP(9),Z(30),P(30),T(30),DEP(30),
     *DZ(30)
      DATA MISS/-32768/
      DOUBLE PRECISION WEIGHT
      DO 1 J=1,NRRL
      Z(J)=FLOAT(IZ(J))*10.
      P(J)=FLOAT(IP(J))*.1
      P(J)=ABS(P(J))
      T(J)=FLOAT(IT(J))*.1
    1 DEP(J)=FLOAT(IDD(J))*.1
      VALU=10.
      MST=0
      MSDD=0
C
C     CHECK FOR MISSING STATION PRESSURE
C
```

I-52


```
          IF(IP(1) .EQ. MISS) GO TO 10
          IEND=1
          GO TO 20
       10 IEND=2
          VALU=9.
       20 LEV1=NRRL
          LEVSTR=0
          LEVEND=0
          LEVHGT=0
C
C         LOOP TO CALCULATE TEMPERATURES FOR INTERMEDIATE PRESSURE LEVELS
C
          DO 60 I=1,NRRL
          LEV=NRRL+1-I
          IF(LEV .LT. IEND) GO TO 65
C
C         TAG LEVEL OF LOWEST HEIGHT AVAILABLE
C
          IF(IZ(LEV) .EQ. MISS) GO TO 30
          LEVHGT=LEV
C
C         JUMP TO 50 IF TEMPERATURE IS MISSING
C
       30 IF(IT(LEV) .EQ. MISS) GO TO 50
          LEV2=LEV1
          LEV1=LEV
C
C         JUMP TO 60 IF NO PREVIOUS MISSING TEMPERATURES
C
          IF(LEVSTR .EQ. 0) GO TO 60
          DELT=T(LEV2)-T(LEV1)
          DLNP=ALOG(P(LEV2)/P(LEV1))
          DO 40 ILEV=LEVEND,LEVSTR
C
C         CALCULATE MISSING TEMPERATURES FOR INTERMEDIATE PRESSURE LEVELS
C               USING LOG PRESSURE INTERPOLATION
C
          T(ILEV)=T(LEV1)+(DELT/DLNP)*ALOG(P(ILEV)/P(LEV1))
       40 CONTINUE
          LEVSTR=0
          LEVEND=0
          GO TO 60
       50 LEVEND=LEV
          MST=MST+1
          IF(LEVSTR .GT. 0) GO TO 60
          LEVSTR=LEV
       60 CONTINUE
C
C         JUMP TO 80 IF NO PREVIOUS MISSING TEMPERATURES
C
       65 IF(LEVSTR .EQ. 0) GO TO 80
          DELT=T(LEV2)-T(LEV1)
          DLNP=ALOG(P(LEV2)/P(LEV1))
          DO 70 ILEV=LEVEND,LEVSTR
C
C         CALCULATE MISSING TEMPERATURES FOR PRESSURE LEVELS NEAR SURFACE
C
       70 T(ILEV)=T(LEV1)+(DELT/DLNP)*ALOG(P(ILEV)/P(LEV1))
C
```

I-53


```
      C     JUMP TO 130 IF STATION PRESSURE IS NOT MISSING
      C
   80 IF(IP(1) .NE. MISS) GO TO 130
      C
      C     JUMP TO 90 IF ANY HEIGHTS OF RAOB REPORTING LEVELS WERE GIVEN
      C
      IF(LEVHGT .GT. 0) GO TO 90
      C
      C     CALCULATE STATION PRESSURE ASSUMING STANDARD PRESSURE FOR STATION
      C          ELEVATION
      C
      P(1)=1013.25*(1.-(Z(1)/44308.))**(5.256794407)
      C
      C     STATION PRESSURE IS THE GREATEST OF STANDARD ATMOSPHERE PRESSURE
      C          AND LOWEST PRESSURE LEVEL
      C
      P(1)=AMAX1(P(1),P(2))
      GO TO 130
      C
      C     JUMP TO 110 IF A HEIGHT IS GIVEN FOR THE LOWEST PRESSURE LEVEL
      C
   90 IF(LEVHGT .LE. 2) GO TO 110
      C
      C     CALCULATE HEIGHTS COMING DOWN FROM LOWEST HEIGHT GIVEN USING
      C          LOG PRESSURE
      C
      ILEV=LEVHGT-2
      DO 100 I=1,ILEV
      LEV=LEVHGT-I
      AVET=0.5*(T(LEV+1)+T(LEV))
  100 Z(LEV)=Z(LEV+1) + 29.2898*AVET*ALOG(P(LEV+1)/P(LEV))

      C     JUMP TO 110 IF HEIGHT OF SECOND RAOB LEVEL IS ABOVE SURFACE.

      IF(Z(2) .GT. Z(1)) GO TO 110
      P(1)=P(2)

      C     STATION PRESSURE SAME AS LOWEST PRESSURE LEVEL, JUMP TO 130

      GO TO 130

      C     TEST FOR MISSING STATION TEMPERATURE, JUMP TO 120 IF SO.

  110 IF(IT(1) .EQ. MISS) GO TO 120
      AVET=0.5*(T(1)+T(2))

      C     CALCULATE STATION PRESSURE WITH NO ASSUMPTIONS.

      P(1)=P(2)*EXP((Z(2)-Z(1))/(29.2898*AVET))
      GO TO 150

      C     CALCULATE STATION PRESSURE USING THE STANDARD ATMOSPHERE
      C     PRESSURE GRADIENT

  120 P(1)=P(2) + .1202141133*(Z(2)-Z(1))*(1.-((Z(1)+Z(2))/88616.))**4.2
     *56794407
      GO TO 140

      C     TEST FOR MISSING STATION TEMPERATURE. JUMP TO 150 IF NOT.
```

CFAS SUBPROGRAM ELEMENT RAOB

```
      130 IF(IT(1) .GT. 0) GO TO 150
      140 MST=MST+1
          DELT=T(LEV2)-T(LEV1)
          DLNP=ALOG(P(LEV2)/P(LEV1))

  C       CALCULATE STATION TEMPERATURE USING LOG PRESSURE

          T(1)=T(LEV1)+(DELT/DLNP)*ALOG(P(1)/P(LEV1))

  C       CALCULATE MISSING DEWPOINT DEPRESSIONS ASSUMING MOTOR
  C             BOATING

      150 DO 160 LEV=1,NRRL
          IF(IDD(LEV) .GE. 0) GO TO 155
          DEP(LEV)=.285*(T(LEV)-273.2)+20.6
          MSDD=MSDD+1
      155 IF(DEP(LEV) .LT. 0.) DEP(LEV)=0.
      160 CONTINUE

  C       CHECK TO SEE THAT LOWEST LEVEL WITH HEIGHT IS THE LOWEST PRESSURE
  C               LEVEL ABOVE THE SURFACE, IF SO JUMP TO 180

          IF(LEVHGT .LE. 2) GO TO 180

  C       WIPE OUT HEIGHTS CREATED PREVIOUSLY

          LEVHGT=LEVHGT-1
          DO 170 LEV=2,LEVHGT
          IZ(LEV)=MISS
      170 Z(LEV)=IZ(LEV)
      180 LEVHGT=1

  C       CALCULATE MISSING HEIGHTS OF PRESSURE LEVELS

          DO 230 LEV=2,NRRL

  C       JUMP TO 190 IF HEIGHT OF PRESSURE LEVEL IS NOT MISSING.

          IF(IZ(LEV) .NE. MISS) GO TO 190

  C       CALCULATE MISSING HEIGHT

          AVET=.5*(T(LEV-1)+T(LEV))
          DZ(LEV-1)=-29.28980*AVET*ALOG(P(LEV)/P(LEV-1))
          Z(LEV)=Z(LEV-1)+DZ(LEV-1)
          GO TO 230

  C        JUMP OUT OF LOOP IF CALCULATED HEIGHT IS ABOVE MIDPOINT
  C           OF HIGHEST CFDB LAYER

      190 IF(Z(LEV) .GE. HMP(9)) GO TO 240
          IF(LEVHGT .LT. (LEV-1)) GO TO 210
      200 LEVHGT=LEV
          GO TO 230
     210  AVET=0.5*(T(LEV-1)+T(LEV))
  C
  C               CALCULATE TEST HEIGHT
  C
```

CFAS SUBPROGRAM ELEMENT RAOB

```
          ZTEST=Z(LEV-1)-29.2898*AVET*ALOG(P(LEV)/P(LEV-1))
C
C         IF TEST HEIGHT NOT EQUAL REPORTED HEIGHT, NORMALIZE
C            TO FIT
C
          IF(ZTEST .EQ. Z(LEV)) GO TO 200
          DELZ=Z(LEV)-Z(LEVHGT)
          DTESTZ=ZTEST-Z(LEVHGT)
          ZNORM=DELZ/DTESTZ
          LEVEND=LEV-1
          LEVSTR=LEVHGT+1
          DO 220 ILEV=LEVSTR,LEVEND
          Z(ILEV)=Z(ILEV-1)+DZ(ILEV-1)*ZNORM
      220 CONTINUE
          GO TO 200
      230 CONTINUE
C
C         CALCULATE TEMPERATURE, PRESSURE AND DEWPOINT DEPRESSION
C         AT THE MIDPOINTS OF THE CFOB LAYERS
C
      240 DO 270 LAY=1,9
          TMP(LAY)=-32768.
          DO 250 LEV=2,NRRL
          IF(Z(LEV) .GE. HMP(LAY)) GO TO 260
      250 CONTINUE
          GO TO 270
      260 WEIGHT=(DBLE(HMP(LAY))-DBLE(Z(LEV-1)))/(DBLE(Z(LEV))
         *-DBLE(Z(LEV-1)))
          DMP(LAY)=DEP(LEV-1)+(DEP(LEV)-DEP(LEV-1))*WEIGHT
          TMP(LAY)=T(LEV-1)+(T(LEV)-T(LEV-1))*WEIGHT
          PMP(LAY)=DBLE(P(LEV-1))*(DBLE(P(LEV))/DBLE(P(LEV-1)))**WEIGHT
      270 CONTINUE

C         CALCULATE VALU OF RAOB.

          XRRL=NRRL
          XMST=MST
          XMSDD=MSDD
          VALU=VALU-((XMST/XRRL)*4.)-((XMSDD/XRRL)*4.)
          RETURN
          END
```

CFAS SUBPROGRAM ELEMENT RETOBR

CLOUD-FOG*CFAS.RETOBR

```
      SUBROUTINE RETOBR(INCODE,NTIME,INOBEL,NOMORE,TYMOLD)
C
C     THIS ROUTINE RETRIEVES AN OBS/REP FROM THE FILE AN CHECKS FOR THE
C         PRESENCE OR PROBABILITY OF CONVECTIVE TYPE CLOUDS.
C
      INTEGER TYMOLD
C
      DIMENSION INOBEL(44)
C
      CALL GETIOW(INCODE,NTIME,INOBEL,NOMORE)
C
C     JUMP TO 70 IF NO MORE OBS/REP IN THE FILE.
C
      IF(NOMORE .EQ. 1) GO TO 70
C
C     SET NOMORE=1 AND JUMP TO 70 IF OBS/REP REMAINING ON THE FILE WERE
C     MADE BEFORE TYMOLD.
C
      IF(NTIME .LT. TYMOLD) GO TO 4
    2 IF(INOBEL(4) .GE. TYMOLD) GO TO 8
      NOMORE=1
      GO TO 70
    4 IF(INOBEL(4) .LE. NTIME) GO TO 8
      GO TO 2
C
C     JUMP TO 70 IF NOT A TYPE 1,2 OR 3 OBS/REP.
C
    8 IF(INOBEL(6) .GT. 3) GO TO 65
C
C     CHECK FOR THE PRESENCE OF CONVECTIVE CLOUDS IN LOW CLOUDS
C
      LT=INOBEL(23)
      IF((LT .LE. 0) .OR. (LT .GT. 9)) GO TO 10
      IF(LT .EQ. 6) GO TO 10
      LT=1 + (10*LT)
C
C     CHECK FOR PRESENCE OF MIDDLE CLOUDS.
C
      MT=INOBEL(25)
      IF((MT .GT. 0) .AND. (MT .LE. 9)) LT=LT+1
C
C     CHECK FOR ABSENCE OF HIGH CLOUDS
C
      IHT=INOBEL(26)
      IF((IHT .LE. 0) .OR. (IHT .GT. 9)) GO TO 60
      LT=LT+1
      GO TO 60
C
C     CHECK LAYERED CLOUD DATA FOR PRESENCE OF CONVECTIVE TYPE CLOUDS
C
   10 LTT=0
      DO 40 ITC=27,36
      LT=INOBEL(ITC)
      IF((LT .LT. 8) .OR. (LT .GT. 9)) GO TO 30
      LTT=-11
      GO TO 40
   30 IF(LTT .EQ. 0) GO TO 40
      IF((LT .LT. 0) .OR. (LT .GT. 9)) GO TO 40
```

CFAS SUBPROGRAM ELEMENT RETOBR

```
          LTT=LTT-1
       40 CONTINUE
          IF(LTT .EQ. 0) GO TO 45
          IF(LTT .LT. -13) LTT=-13
          LT=LTT
          GO TO 60
C
C         CHECK WEATHER FOR PROBABILITY OF CONVECTIVE TYPE CLOUDS
C
       45 DO 50 ITC=37,43
          IWT=MOD(INOBEL(ITC),100)
          IF((IWT .LT. 17) .OR. (IWT .GT. 99)) GO TO 50
          IF((IWT .GE. 30) .AND. (IWT .LE. 79)) GO TO 50
          IF((IWT .GE. 20) .AND. (IWT .LE. 24)) GO TO 50
          IF(IWT .EQ. 28) GO TO 50
          LT=-22
          GO TO 60
       50 CONTINUE
          IWT=INOBEL(44)
          IF((IWT .LT. 8) .OR. (IWT .GT. 9)) GO TO 55
          LT=-22
          GO TO 60
       55 LT=3
       60 INOBEL(23)=LT
          GO TO 70
       65 INOBEL(23)=-32768
       70 RETURN
          END
```

CFAS SUBPROGRAM ELEMENT SECTOR

```
CLOUD-FOG*CFAS.SECTOR
          SUBROUTINE SECTOR
C ESTABLISH THE STORAGE SECTOR MAP
          COMMON /BASE/ DXSECT, DYSECT, EDGE, IBLOCK, IDTIME, IDXUTM,
         * IDYUTM, INUMBR, ISTATI, ISTATO, JNUMBR, JSTATI, JSTATO, JTIME,
         * LASTJ, MAXGPS, NBJNOW, NBLKFJ, NCOLS, NGX, NGY, NINI, NINTAB,
         * NROWS, NRPBFI, NRPBFJ, NSECTR, NWDBKI, NWDBKJ, NWDREC, NXSECT,
         * NYSECT, UTMPGD, XBASE, XMAX, XMIN, YBASE, YMAX, YMIN,
         * NNEWRS(100), NALLRS(100), ITABLE(4, 500), IBUF(3750), JBUF(1000),
         * JTIMES(100), IRXMAX, IRXMIN, IRYMAX, IRYMIN
C                 NGX AND NGY ARE THE NO. OF GRID POINTS CONTAINED IN THE X
C                 AND Y DIRECTIONS OF A STORAGE SECTOR.
          NGX=1
       10 IF (NGX*NGX .GT. MAXGPS) GO TO 20
          NGX=NGX+1
          GO TO 10
       20 NGX=NGX-1
C                 NOTE - SQUARE STORAGE SECTORS ARE BEING USED.  AT SOME
C                 FUTURE TIME IT MAY BE ADVANTAGEOUS IN TERMS OF OPERATING
C                 EFFICIENCY TO USE RECTANGULAR STORAGE SECTORS.
          NGY=NGX
C                 DEFINE STORAGE SECTOR DIMENSIONS IN HECTOMETERS.
          DXSECT=NGX*UTMPGD
          DYSECT=DXSECT
C                 ALIGNMENT IN THE X DIRECTION.
C                 MINIMUM LEFT POSITION OF LEFT EDGE OF LEFTMOST STORAGE
C                 SECTOR IN HECTOMETERS.
          XLEFT=XBASE-EDGE
C                 MINIMUM RIGHT POSITION OF RIGHT EDGE OF RIGHTMOST STORAGE
C                 SECTOR IN HECTOMETERS
          XRIGHT=XBASE+(NCOLS-1)*UTMPGD+EDGE
C                 ILEFT = INTEGER NO. OF EAST-WEST GRID POINTS WHICH WOULD BE
C                 REQUIRED TO COVER THE DEFINED EDGE DISTANCE.
          ILEFT=EDGE/UTMPGD
C                 XMIN = UTM UNITS OF LEFT EDGE OF LEFTMOST SECTOR SUCH THAT
C                 STORAGE SECTOR BOUNDARIES WILL FALL HALF WAY BETWEEN GRID
C                 POINTS.
       30 XMIN=XBASE-(ILEFT-0.5)*UTMPGD
          IF (XMIN .LE. XLEFT) GO TO 40
          ILEFT=ILEFT+1
          GO TO 30
C                 NXSECT = INTEGER NO. OF EAST-WEST STORAGE SECTORS REQUIRED
C                 TO COVER SPACE FROM XMIN TO XRIGHT.
       40 NXSECT=(XRIGHT-XMIN)/DXSECT
       50 XMAX=XMIN+NXSECT*DXSECT
          XDIF=XMAX-XRIGHT
          IF (XDIF) 60, 80, 70
       60 NXSECT=NXSECT+1
          GO TO 50
C                 XDIF = EXCESS DISTANCE ON RIGHT SIDE OF RIGHT EDGE.
C                 CONVERT THIS DISTANCE TO GRID UNITS AND TRY TO SPLIT IT UP
C                 ON BOTH SIDES BY COMPUTING THE NO. OF GRID UNITS TO MOVE IN
C                 THE LEFT X DIRECTION.
       70 NGMOVX=0.5*XDIF/UTMPGD
          XMIN=XBASE-(ILEFT+NGMOVX-0.5)*UTMPGD
          XMAX=XMIN+NXSECT*DXSECT
       80 IRXMAX=XMAX-XBASE
          IRXMIN=XBASE-XMIN
          IRXMIN=-IRXMIN
```

CFAS SUBPROGRAM ELEMENT SECTOR

```
C         ALIGNMENT IN THE Y DIRECTION IS DONE IN THE SAME MANNER AS
C         ALIGNMENT IN THE X DIRECTION.
      YBOT=YBASE-EDGE
      YTOP=YBASE+(NROWS-1)*UTMPGD+EDGE
      IDOWN=EDGE/UTMPGD
   90 YMIN=YBASE-(IDOWN-0.5)*UTMPGD
      IF (YMIN .LE. YBOT) GO TO 100
      IDOWN=IDOWN+1
      GO TO 90
  100 NYSECT=(YTOP-YMIN)/DYSECT
  110 YMAX=YMIN+NYSECT*DYSECT
      YDIF=YMAX-YTOP
      IF (YDIF) 120, 140, 130
  120 NYSECT=NYSECT+1
      GO TO 110
  130 NGMOVY=0.5*YDIF/UTMPGD
      YMIN=YBASE-(IDOWN+NGMOVY-0.5)*UTMPGD
      YMAX=YMIN+NYSECT*DYSECT
  140 IRYMAX=YMAX-YBASE
      IRYMIN=YBASE-YMIN
      IRYMIN=-IRYMIN
      NSECTR=NXSECT*NYSECT
      PRINT 400 NSECTR
  400 FORMAT (1H , ' SECTOR -', I4, ' STORAGE SECTORS WILL BE USED FOR S
     *TORAGE OF RECENT OBS/REP DATA RECORDS IN FILE I.')
      PRINT 410 DXSECT, DYSECT
  410 FORMAT (1H , ' SECTOR - EACH STORAGE SECTOR COVERS', F6.1, ' HECTO
     *METERS IN THE X DIRECTION AND', F6.1, ' HECTOMETERS IN THE Y DIREC
     *TION')
      XMINK=XMIN/10.0
      XMAXK=XMAX/10.0
      PRINT 420 NXSECT, IRXMIN, IRXMAX, XMINK, XMAXK
  420 FORMAT (1H , ' SECTOR -', I3, ' STORAGE SECTORS IN THE X DIRECTION
     * WILL SPAN RELATIVE X COORDINATES', I9, '  THROUGH', I9, '  HECTOM
     *ETERS', /, 44X, 'REPRESENTING ABSOLUTE UTM COORDINATES', F9.2, '
     *THROUGH', F9.2, '  KILOMETERS')
      YMINK=YMIN/10.0
      YMAXK=YMAX/10.0
      PRINT 430 NYSECT, IRYMIN, IRYMAX, YMINK, YMAXK
  430 FORMAT (1H , ' SECTOR -', I3, ' STORAGE SECTORS IN THE Y DIRECTION
     * WILL SPAN RELATIVE Y COORDINATES', I9, '  THROUGH', I9, '  HECTOM
     *ETERS', /, 44X, 'REPRESENTING ABSOLUTE UTM COORDINATES', F9.2, '
     *THROUGH', F9.2, '  KILOMETERS')
      RETURN
      END
```

@HDG,P CFAS SUBPROGRAM ELEMENT SFDINT

@PRT,S CFAS.SFDINT
FURPUR C026-10/28-13:58

CFAS SUBPROGRAM ELEMENT SFDINT

CLOUD-FOG*CFAS.SFDINT

```
      SUBROUTINE SFDINT
C
C     ROUTINE TO INTERPRET SURFACE OBS/REP IN TERMS OF CFDB PARAMETERS.
C
C     SOURCES OF INPUT DATA ARE AVIATION WEATHER REPORTS IN AIRWAYS AND
C       METAR CODES AND SURFACE SYNOPTIC REPORTS IN SYNOP CODE
C
C     INPUT DATA
C
C  IX = X DISTANCE OF OBS/REP SITE FROM IXREF, HECTOMETERS
C  IY = Y DISTANCE OF OBS/REP SITE FROM IYREF, HECTOMETERS
C  IZ = TERRAIN HEIGHT AT OBS/REP SITE, METERS
C  ITIME = TIME OF OBS/REP
C  ITYPE = TYPE OF OBS/REP
C              1=AIRWAYS  -1 IF A SPECIAL
C              2=METAR      -2 IF A SPECI (SPECIAL)
C              3=SYNOP
C  IDD = WIND DIRECTION, 0-360 FROM TRUE NORTH
C  IFF = WIND SPEED, METERS/SEC.
C  IPPP = SEA LEVEL PRESSURE, MILLIBARS
C  ITT = SURFACE TEMPERATURE, DEGREES KELVIN
C  ITD= SURFACE DEWPOINT, DEGREES KELVIN
C  ITSC = TOTAL SKY COVER, 0-9   WMO CODE 2700
C  IVIS = VISIBILITY-
C                 AIRWAYS - STATUTE MILES*10000
C                METAR - METERS
C                SYNOP - WMO CODE 4377
C  NWEA(J) = PRESENT WEATHER-- FROM 1 TO 7 ELEMENTS MAY BE INPUT
C                AIRWAYS - CFAS CODE 1
C                METAR - WMO CODE 4678
C                SYNOP - WMO CODE 4677
C  IPW = PAST WEATHER, 0-9  WMO CODE 4500
C  NH = SKY COVER DUE TO LOW OR MIDDLE CLOUDS, 0-9   WMO CODE 2700
C  ICL = LOW CLOUD TYPE, 0-9   WMO CODE 0513
C  IH = HEIGHT ABOVE GROUND OF LOWEST CLOUD, 0-9   WMO CODE 1600
C  ICM = MIDDLE CLOUD TYPE, 0-9   WMO CODE 0515
C  ICH = HIGH CLOUD TYPE, 0-9   WMO CODE 0509
C  NS(J) = SKY COVER DUE TO CLOUD LAYER - FROM 1 TO 10 LAYERS
C                AIRWAYS - CFAS CODE 2
C                METAR - WMO CODE 2700
C                SYNOP - WMO CODE 2700
C  ICTS(J) = TYPE OF CLOUD IN LAYER, 0-9   WMO CODE 0500
C  IHS(J) = HEIGHT OF BASE OF CLOUD LAYER
C                AIRWAYS - 100'S OF FEET
C                METAR - WMO CODE 1677
C                SYNOP - WMO CODE 1677
C  ITHN(J) = CLOUD LAYER THICKNESS INDICATOR
C                1 IF THIN
C                MISSING IF NOT THIN
C  ICLG = CEILING DESIGNATOR - FIRST TWO DIGITS ARE THE INDEX NO. J OF
C     THE CEILING LAYER.  THIRD DIGIT HAS A FOLLOWING MEANING
C                  1 = MEASURED
C                  2 = AIRCRAFT
C                  3 = BALLOON
C                  4 = RADAR
C                  5 = ESTIMATED
C                  6 = INDEFINITE
C  ICLGV = CHARACTERISTIC OF CEILING
```

CFAS SUBPROGRAM ELEMENT SFDINT

```
      C                    MISSING = NOT VARIABLE
      C                     1 = VARIABLE
      C  IVISC = VISIBILITY CHARACTERISTICS
      C                   MISSING = NOT VARIABLE
      C                   1 = VARIABLE
      C
      C  CLOUD/FOG DATA BASE PARAMETERS
      C
      C  IVALU = INFORMATION VALUE OF THE OBS/REP (1-10)
      C          0 INDICATES NO DATA USEABLE FOR DETERMINING ANY CFDB PARAMS.
      C          10 INDICATES AN OBS/REP WITH ALL NEEDED DATA PRESENT AND
      C                   USEABLE.
      C          1 TO 9 INDICATES AN OBS/REP WITH SOME MISSING OR NON-USEABLE
      C                   DATA.
      C  NTCLC = TOTAL CLOUD COVER. (00 - 100)
      C  NCEIL = HEIGHT OF CEILING LAYER (AGL), DEKAMETERS + TYPE OF CEILING
      C                   DIGIT AS PER THIRD DIGIT OF ICLG.  MINUS IF VARIABLE.
      C  MINBAS = HEIGHT OF BASE OF LOWEST CLOUD (AGL), DEKAMETERS.
      C  MAXTOP = HEIGHT OF THE TOP OF HIGHEST CLOUD (ACL), DEKAMETERS.
      C  MSPWE = MOST SIGNIFICANT PRESENT WEATHER ELEMENT (WMO CODE 4677)
      C  NVV = PREVAILING VISIBILITY AT SURFACE, METERS. NEGATIVE IF VARIABLE.
      C  LCOV(9) = PERCENT CLOUD COVER IN THE CFDB LAYERS
      C
      C DERIVED LAYERED CLOUD INFORMATION
      C
      C  NUMLAY = NUMBER OF LAYERS GENERATED
      C  KIND = KIND OF CLOUD LAYER
      C                   1 = LOW
      C                   2 = MIDDLE
      C                   3 = HIGH
      C                   4 = FOG
      C                   5 = LOWEST CLOUD
      C                   6 = CLEAR LAYER
      C  ITHIN = THIN LAYER DESIGNATOR
      C                   MISSING = NOT THIN
      C                   1 = THIN
      C  COVER = CLOUD COVER IN LAYER (0.0 - 1.0)
      C  BASE = HEIGHT OF THE BASE OF LAYER, FEET.
      C  TOP = HEIGHT OF TOP OF CLOUD LAYER, FEET.
      C
      C MAP AND WINDOW DATA
      C
      C  XREF=EAST-WEST UTM GRID COORDINATE OF LOWER LEFT HAND CORNER OF THE
      C                   WINDOW, KM.
      C  YREF= NORTH-SOUTH UTM GRID COORDINATE OF LOWER LEFT HAND CORNER OF
      C                   THE WINDOW, KM.
      C  CMRD = CENTRAL MERIDIAN OF WINDOW


        COMMON /OBSREP/ IX,IY,IZ,ITIME,IOBC,ITYPE,IVALU,NTCLC,NCEIL,NVV,
       *MINBAS,MAXTOP,MSPWE,LCOV(9),ICL,ITSC,ICM,ICH,ICTS(10),NWEA(7),IPW,
       *IDD,IFF,IPPP,ITT,ITD,IVIS,NH,IH,NS(10),IHS(10),ITHN(10),ICLG,ICLGV
       *,IVISC,NOUSE(58)

        COMMON/CLOUDS/NUMLAY,KIND(10),ITHIN(10),COVER(10),BASE(10),TOP(10)

        DATA MISS/-32768/,FMISS/-32768./

        COMMON/MAP/XREF,YREF,CMRD
```

I-62

CFAS SUBPROGRAM ELEMENT SFDINT

```
              DIMENSION CODE(10)

              DATA CODE/82.,246.,492.,820.,1447.,2620.,4100.,5740.,7380.,-32768.
             */

      C       TOPCLR=ASSUMED TOP OF ALL CLOUDS

              TOPCLR=40000.


      C       INITIALIZE PARAMETERS

              VALU=0.
              MT=IABS(ITYPE)

      C       JUMP TO 480 IF OBS/REP TYPE IS NOT AN AIRWAYS, METAR OR SYNOP.

              IF(MT .GT. 3) GO TO 480
              NUMLAY=0
              DO 10 I=1,10
              KIND(I)=MISS
              ITHIN(I)=MISS
              COVER(I)=FMISS
              BASE(I)=FMISS
           10 TOP(I)=FMISS
              NTCLC=MISS
              NCEIL=MISS
              MINBAS=MISS
              MAXTOP=MISS
              MSPWE=-1
              NVV=MISS
              DO 20 I=1,9
           20 LCOV(I)=MISS

      C       CALCULATE LATITUDE OF OBS/REP.

              XUTM=IX
              XUTM=(XREF+XUTM/10.)/100.
              YUTM=IY
              YUTM=(YREF+YUTM/10.)/100.
              CALL BAKUTM(DLONG,DLAT,XUTM,YUTM,CMRD)

      C       CONSTRUCT CLOUD LAYERS FROM LAYER CLOUD DATA IF PRESENT

              IF(NS(1) .GE. 0) CALL LAYCLD(DLAT,VALU)

      C       CONVERT IH OF SYNOP CODE TO FEET

              IF(IH .GT. 8 .OR. IH .LT. 0) GO TO 110
              HITLOW=CODE(IH+1)
              GO TO 120
          110 HITLOW=FMISS

      C       DETERMINE MOST SIGNIFICANT PRESENT WEATHER ELEMENT.

          120 DO 130 IW=1,7
              IF(NWEA(IW) .LT. 0) GO TO 128
              IF(NWEA(IW) .GT. 99 .AND. MT .NE. 1) GO TO 128
```

CFAS SUBPROGRAM ELEMENT SFDINT

```
          MNWEA=MOD(NWEA(IW),100)
          MMSPW=MOD(MSPWE,100)
          IF(MNWEA-MMSPW) 126,122,124
      122 MSPWE=MAXO(NWEA(IW),MSPWE)
          GO TO 126
      124 MSPWE=NWEA(IW)
      126 NWEA(IW)=MNWEA
          GO TO 130
      128 NWEA(IW)=MISS
      130 CONTINUE

C     JUMP TO 165 IF VISIBILITY IS MISSING

          IF(IVIS .LT. 0) GO TO 165

C         CONVERT AIRWAYS AND SYNOP VISIBILITY CODES TO VISIBILITY IN METERS

          GO TO (140,160,150),MT

C         AIRWAYS CODE CONVERSION

      140 VIS=IVIS
          VIS=VIS*0.16093
          IVIS=VIS
          GO TO 160

C         SYNOP CODE CONVERSION

      150 IF(IVIS .GT. 50) GO TO 152
          IVIS=IVIS*100
          GO TO 160
      152 IF(IVIS .GT. 80) GO TO 154
          IVIS=(IVIS-50)*1000
          GO TO 160
      154 IF(IVIS .LE. 89) GO TO 156
          IVIS=MISS
          GO TO 160
      156 IVIS=32760
      160 NVV=IVIS

C         MAKE NVV NEGATIVE IF VISIBILITY IS VARIABLE

          IF(IVISC .EQ. 1) NVV=-NVV

C         JUMP TO 170 IF THERE WAS NO LAYERED CLOUD DATA IN THE OBS/REP

      165 IF(NUMLAY .EQ. 0) GO TO 170

C         CHECK FOR FOG AND ESTIMATE PERCENTAGE CLOUD COVER AND TOPS OF
C            CLOUD LAYERS FROM HORIZONTAL VISIBILITY AND TYPE OF FOG

          CALL FOG(IVIS,NWEA,AMT,VALU)

C         JUMP IF LOWEST CLOUD HEIGHT IS MISSING

          IF(HITLOW .EQ. FMISS) GO TO 220

C         CODE A 1/16 CLOUD COVER

```

```
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=5
            COVER(NUMLAY)=0.0625
            BASE(NUMLAY)=HITLOW
            GO TO 225

C     CALCULATE TOTAL SKY COVER FROM CODE IF NOT MISSING

  170 IF(ITSC .LT. 0 .OR. ITSC .GT. 8) GO TO 180
      CTOT=ITSC/8.

C     ASSURE LOW-MIDDLE CLOUD COVER NOT GREATER THAN TOTAL SKY COVER
C     WHEN TOTAL SKY COVER NOT MISSING OR OBSCURED

      IF(NH .GT. ITSC .AND. NH .LE. 9) NH=ITSC
      GO TO 190
  180 CTOT=FMISS

C     JUMP IF LOWEST CLOUD AMOUNT PRESENT

  190 IF(NH .GE. 0 .AND. NH .LE. 9) GO TO 200
      CLOW=FMISS
      GO TO 210

C     TREAT OBSCURED LOWEST CLOUD AMOUNT AS OVERCAST

  200 IF(NH .EQ. 9) NH=8
      CLOW=NH/8.

C     CHECK FOR FOG AND ESTIMATE PERCENTAGE CLOUD COVER AND TOPS OF
C        CLOUD LAYERS FROM HORIZONTAL VISIBILITY AND TYPE OF FOG

  210 CALL FOG(IVIS,NWEA,AMT,VALU)

C     JUMP IF FOG COMPLETELY COVERS SKY

      IF(NUMLAY .GT. 0 .AND. AMT .GT. .99) GO TO 225

C     CONSTRUCT CLOUD LAYERS FROM MANDATORY SYNOP TYPE DATA

      IF(ICL .GT. 9) ICL=MISS
      IF(ICM .GT. 9) ICM=MISS
      IF(ICH .GT. 9) ICH=MISS
      CALL SYNOP(CTOT,CLOW,HITLOW,ICL,ICM,ICH,NWEA,DLAT,VAL,MSPWE)
      VALU=(VALU+VAL)/2.

C     IF NO LAYERED CLOUD INFORMATION OBTAINABLE FROM OBS/REP, JUMP TO
C          490

  220 IF(NUMLAY .EQ. 0) GO TO 490

C     JUMP IF LOWEST CLOUD BASE IS MISSING

  225 IF(HITLOW .LE. 0) GO TO 300

C     DETERMINE LOCATION OF THE LOWEST CLOUD

      DO 230 LNO=1,NUMLAY
      IF(KIND(LNO) .EQ. 5) GO TO 240
```

CFAS SUBPROGRAM ELEMENT SFCINT

```
    230 CONTINUE
C       DETERMINE CLOUD COVER FOR LOWEST BASE.

    240 DO 260 LNX=1,NUMLAY
        IF(KIND(LNX) .EQ. 1) GO TO 250
        IF(KIND(LNX) .NE. 2) GO TO 260
    250 CLDINT=-0.0714285714 + 1.07142857*COVER(LNX)
        COVER(LNO)=AMAX1(CLDINT,0.0625)
        GO TO 300
    260 CONTINUE

C       DETERMINE CLOUD TOPS

    300 ELEV=IZ*3.2808

        CALL TOPS(ELEV,NWEA,DLAT)

C       LOWER THE HEIGHTS OF THE TOPS OF LAYERS DESIGNATED AS THIN

        DO 320 LNX=1,NUMLAY
        LTYP=KIND(LNX)
        GO TO (310,310,320,320,320,320),LTYP
    310 IF(ITHIN(LNX) .NE. 1) GO TO 320
        TOP(LNX)=BASE(LNX) + 0.5*(TOP(LNX)-BASE(LNX))
    320 CONTINUE

C       DETERMINE MINBAS AND MAXTOP OF CLOUDS
        BASINT=TOPCLR
        TOPINT=0.

        DO 340 LNX=1,NUMLAY
        LTYP=KIND(LNX)
        IF(LTYP .EQ. 6) GO TO 340
        IF(COVER(LNX) .GE. .025) GO TO 330
        COVER(LNX)=0.05
    330 BASINT=AMIN1(BASINT,BASE(LNX))
        TOPINT=AMAX1(TOPINT,TOP(LNX))
    340 CONTINUE
        MINBAS=BASINT*.03048 + .5
        MAXTOP=TOPINT*.03048 +.5

C       DETERMINE PERCENT CLOUD COVER IN THE CFDB LAYERS AND IDENTIFY
C           LAYERS CONTAINING CLOUDS OBSERVED TO BE THIN

        DO 440 JM=1,4
        DO 430 LNX=1,NUMLAY
        LTYP=KIND(LNX)
        GO TO (360,370,380,390),JM
    360 IF(LTYP .EQ. 6) GO TO 400
        GO TO 430
    370 IF(LTYP .EQ. 5) GO TO 400
        GO TO 430
    380 IF(LTYP .EQ. 4) GO TO 400
        GO TO 430
    390 IF(LTYP .LE. 3) GO TO 400
        GO TO 430
    400 NTBASE=BASE(LNX)
        NTTOP=TOP(LNX)
C       CALCULATE PERCENT CLOUD COVER TO NEAREST 5 PERCENT
```

CFAS SUBPROGRAM ELEMENT SFDINT

```

      NAMT=COVER(LNX)*100. + 2.5
      NAMT=IABS(NAMT-MOD(NAMT,5))
      IF(NAMT .EQ. 0 .AND. KIND(LNX) .NE. 6) GO TO 430

C     IF OBS/REP INDICATED A THIN CLOUD, CODE LAYER WITH A THIN DESIG.

      IF(ITHIN(LNX) .NE. 1) GO TO 410
      NAMT=NAMT+1

C     DETERMINE INDEX NOS. OF LOWEST AND HIGHEST CFDB LAYERS INFLUENCED
C        BY CLOUD LAYER NO. LNX

  410 CALL CFLAY(NTBASE,NTTOP,NTBASE,NTTOP)

      IF(NTBASE .EQ. 0) GO TO 430

C     CODE THE AFFECTED CFDB LAYERS WITH THE PERCENT CLOUD COVER IN
C         CLOUD LAYER NO. LNX

      DO 420 LAY=NTBASE,NTTOP
  420 LCOV(LAY)=NAMT
  430 CONTINUE
  440 CONTINUE

      IF(ITSC .LT. 0 .OR. ITSC .GT. 9) GO TO 450

      IF(ITSC .EQ. 9) ITSC=8

      NTCLC=100.0*(AMT + (1.-AMT)*ITSC/8.0)+ 0.5
      GO TO 460

C     JUMP TO 460 IF NOT A SYNOP TYPE OBS/REP OR TOTAL SKY COVER WAS
C         NOT MISSING IF A SYNOP TYPE OBS/REP

  450 IF(MT .NE. 3) GO TO 460

C     REDUCE VALU TO 5. WHEN TOTAL SKY COVER IS MISSING

      IF(VALU .GT. 5.) VALU=5.

C     JUMP TO 480 IF NO CEILING LAYER

  460 IF(ICLG .LT. 0) GO TO 480
      LSC=ICLG/10
      CEILH=IHS(LSC)*100
      IF(MT .EQ. 1) GO TO 470
      IF(IHS(LSC) .LE. 50) GO TO 470
      CEILH=(IHS(LSC)-50)*1000
      IF(IHS(LSC) .LE. 80) GO TO 470
      CEILH=35000. - (13000./90.)*ABS(DLAT)
  470 NCEIL=CEILH*.03048
      NCEIL=10*NCEIL + MOD(ICLG,10)
      IF(ICLGV .EQ. 1) NCEIL=-NCEIL

  480 IVALU=VALU
      IF(MSPWE .EQ. -1) MSPWE=MISS
      RETURN
  490 IF(NVV .EQ. MISS) GO TO 500
```

CFAS SUBPROGRAM ELEMENT SFDINT

```
          VALU=1.
      500 IF(MSPWE .EQ. -1) GO TO 480
          VALU=VALU+1.
          GO TO 480
          END
```

@HDG,P CFAS SUBPROGRAM ELEMENT STOREC

@PRT,S CFAS.STOREC
FURPUR 0026-10/29-13:58

CFAS SUBPROGRAM ELEMENT STOREC

CLOUD-FOG*CFAS.STOREC

```
          SUBROUTINE STOREC (IREC)
C STORES AN OBS/REP IN THE OBS/REP DATA BASE.
C IREC    = STARTING ADDRESS OF OBS/REP FROM CALLING ROUTINE.
          COMMON /BASE/ DXSECT, DYSECT, EDGE, IBLOCK, IDTIME, IDXUTM,
         * IDYUTM, INUMBR, ISTATI, ISTATO, JNUMBR, JSTATI, JSTATO, JTIME,
         * LASTJ, MAXSPS, NBJNOW, NBLKFJ, NCOLS, NGX, NGY, NINI, NINTAB,
         * NROWS, NRPBFI, NRPBFJ, NSECTR, NWDBKI, NWDBKJ, NWDREC, NXSECT,
         * NYSECT, UTMPGD, XBASE, XMAX, XMIN, YBASE, YMAX, YMIN,
         * NNEWRS(100), NALLRS(100), ITABLE(4, 500), IBUF(3750), JBUF(1000),
         * JTIMES(100), IRXMAX, IRXMIN, IRYMAX, IRYMIN
          DIMENSION IREC(1)
          MYTIME=IREC(IDTIME)
          MYX=IREC(IDXUTM)
          MYY=IREC(IDYUTM)
          IF (MYX .LE. IRXMAX) GO TO 6
        4 PRINT 5 MYTIME, MYX, MYY
        5 FORMAT (1H0, ' STOREC - DATA RECORD RECEIVED WAS TOO DISTANT FOR S
         *TORAGE     TIME =', I5, '     X =', I7, '     Y =', I7, /)
          RETURN
        6 IF (MYX .LT. IRXMIN) GO TO 4
          IF (MYY .GT. IRYMAX) GO TO 4
          IF (MYY .LT. IRYMIN) GO TO 4
          MYSECT=NOSECT(MYX, MYY)
          IF (NINI .LT. NINTAB) GO TO 20
       10 CALL ITOJ
       20 NNEW=NNEWRS(MYSECT)
          IF (NNEW .EQ. NRPBFI) GO TO 10
          IF (NBJNOW .EQ. 0) GO TO 40
          IF (ITMDIF(MYTIME, JTIME) .GE. 0) GO TO 40
          PRINT 30, MYTIME, MYX, MYY
       30 FORMAT (1H0, ' STOREC - DATA RECORD RECEIVED TOO LATE FOR STORAGE
         *     TIME =', I5, '     X =', I7, '     Y =', I7, /)
          RETURN
       40 NALL=NALLRS(MYSECT)
          IF (NALL .EQ. 0) GO TO 50
          IF (MYSECT .EQ. IBLOCK) GO TO 50
          CALL BLKIN (NWDBKI, IBUF, MYSECT, INUMBR, ISTATI)
       50 ICOUNT=0
          J=1
          IF (NINI .EQ. 0) GO TO 100
          DO 60 J=1, NINI
          IF (ITMDIF(MYTIME, ITABLE(1, J)) .GE. 0) GO TO 70
          IF (ITABLE(4, J)/100 .EQ. MYSECT) ICOUNT=ICOUNT+1
       60 CONTINUE
          J=NINI+1
          CO TO 100
       70 JNOW=NINI
       80 DO 90 I=1, 3
       90 ITABLE(I, JNOW+1)=ITABLE(I, JNOW)
          IBKREC=ITABLE(4, JNOW)
          IF (IBKREC/100 .EQ. MYSECT) IBKREC=IBKREC+1
          ITABLE(4, JNOW+1)=IBKREC
          JNOW=JNOW-1
          IF (JNOW .GE. J) GO TO 80
      100 MYREC=ICOUNT+1
          ITABLE(1, J)=MYTIME
          ITABLE(2, J)=MYX
          ITABLE(3, J)=MYY
```

CFAS SUBPROGRAM ELEMENT STOREC

```
       ITABLE(4, J)=MYSECT*100+MYREC
       NINI=NINI+1
       IF (NALL .EQ. NRPBFI) NALL=NRPBFI-1
       MYWORD=(MYREC-1)*NWDREC
       IF (NALL-MYREC .LT. 0) GO TO 120
       NOWGET=NALL*NWDREC
       NOWPUT=NOWGET+NWDREC
  110  IBUF(NOWPUT)=IBUF(NOWGET)
       NOWGET=NOWGET-1
       NOWPUT=NOWPUT-1
       IF (NOWGET .GT. MYWORD) GO TO 110
  120  DO 130 I=1, NWDREC
       MYWORD=MYWORD+1
  130  IBUF(MYWORD)=IREC(I)
       CALL BLKOUT (NWDBKI, IBUF, MYSECT, INUMBR, ISTATO)
       IBLOCK=MYSECT
       NNEWRS(MYSECT)=NNEW+1
       NALLRS(MYSECT)=NALL+1
       RETURN
       END
```

@HDG,P CFAS SUBPROGRAM ELEMENT SYNOP

@PRT,S CFAS.SYNOP
FURPUR 0026-10/28-13:58

CFAS SUBPROGRAM ELEMENT SYNOP

CLOUD-FOG*CFAS.SYNOP

```
          SUBROUTINE SYNOP(CTOT,CLOW,HLOW,LOWT,MIDT,NHIT,NWEA,DLAT,VAL,MSPW)

      C   ROUTINE TO CONVERT TOTAL CLOUD COVER,LOWEST CLOUD COVER,LOWEST
      C      BASE,AND CLOUD TYPES INTO LAYERED CLOUD INFORMATION.

      C   CTOT = TOTAL CLOUD COVER (RANGE 0 - 1)
      C   CLOW = LOWEST CLOUD COVER (RANGE 0 - 1)
      C   HLOW = LOWEST CLOUD BASE IN FEET
      C   LOWT = LOW CLOUD TYPE
      C   MIDT = MIDDLE CLOUD TYPE
      C   NHIT = HIGH CLOUD TYPE
      C   NWEA = PRESENT WEATHER
      C   DLAT = LATITUDE
      C
      C DERIVED LAYERED CLOUD INFORMATION
      C
      C  NUMLAY = NUMBER OF LAYERS GENERATED
      C  KIND = KIND OF CLOUD LAYER
      C              1 = LOW
      C              2 = MIDDLE
      C              3 = HIGH
      C              4 = FOG
      C              5 = LOWEST CLOUD
      C              6 = CLEAR LAYER
      C  ITHIN = THIN LAYER DESIGNATOR
      C              MISSING = NOT THIN
      C              1 = THIN
      C  COVER = CLOUD COVER IN LAYER (0.0 - 1.0)
      C  BASE = HEIGHT OF THE BASE OF LAYER, FEET.
      C  TOP = HEIGHT OF TOP OF CLOUD LAYER, FEET.
      C
          DIMENSION KCURW(5),KPWEA(20),NWEA(7)
          COMMON/CLOUDS/NUMLAY,KIND(10),ITHIN(10),COVER(10),BASE(10),TOP(10)
          DATA
         *BASMID/11700./
         *TOPCLR/40000./
         *REDUCE/0.8/
         *KCURW/1,2,2,2,3/
         *KPWEA/1,4*0,1,3*2,3,3*1,2,1,3*2,0,2/

      C   CALCULATE VALUE ON BASIS OF COMBINATIONS OF MISSING DATA.

          VAL=10.
          IF(CTOT .LT. -1. .AND. CLOW .LT. -1.) VAL=VAL-9.
          IF ((LOWT .GT. 0 .OR. MIDT .GT. 0).AND.(CLOW .LT. -1.)) VAL=VAL-3.
          IF ((LOWT .GT. 0 .OR. MIDT .GT. 0 .OR. CLOW .GT. 0.0001) .AND.
         * (HLOW .LT. -1.0)) VAL=VAL-3.0
          IF(CTOT .GT. .0001 .AND. LOWT .LE. 0 .AND. MIDT .LE. 0 .AND.
         *    NHIT .LE. 0) VAL=VAL-2.
          IF(VAL .LT. 0.) VAL=0.

      C   CALCULATE ASSUMED HIGH CLOUD BASE.

          BASHI=35000.-(13000./90.)*ABS(DLAT)

      C   CALCULATE ASSUMED LOW CLOUD BASE,

          KWEA=0
```

CFAS SUBPROGRAM ELEMENT SYNOP

```
         DO 20 NUMWEA=1,7
         IF (NWEA(NUMWEA) .LT. 10) GO TO 20
         IF (NWEA(NUMWEA) .LT. 50) GO TO 10
         INDEX=NWEA(NUMWEA)/10-4
         KWEA=MAX0(KWEA,KCURW(INDEX))
         GOTO20
      10 IF (NWEA(NUMWEA) .GT. 29) GO TO 20
         INDEX=NWEA(NUMWEA)-9
         KWEA=MAX0(KWEA,KPWEA(INDEX))
      20 CONTINUE
         BASLOW=2200.-300.*KWEA
   C     SET INDICATOR FOR NO CB OR TCU.
         NCB=1
   C     JUMP IF LOWEST BASE IS MISSING.
         IF (HLOW .LE. 0.0) GO TO 30
   C     CODE 1/16 CLOUD COVER
         NUMLAY=NUMLAY+1
         KIND(NUMLAY)=5
         COVER(NUMLAY)=0.0625
         BASE(NUMLAY)=HLOW
   C     JUMP IF TOTAL CLOUD COVER NOT MISSING AND NOT ZERO
      30 IF (CTOT .GE. 0.05) GO TO 110
   C     JUMP IF TOTAL CLOUD COVER ZERO.
         IF (ABS(CTOT) .LE. 0.00001) GO TO 100
   C     JUMP IF LOWEST CLOUD COVER NOT MISSING OR ZERO.
         IF (CLOW .GE. 0.05) GO TO 40
   C     RETURN IF LOWEST CLOUD COVER MISSING.
         IF (CLOW .GT. -1.0) GO TO 35
         RETURN
   C     CODE LOW CLEAR
      35 NUMLAY=NUMLAY+1
         KIND(NUMLAY)=6
         COVER(NUMLAY)=0.
         BASE(NUMLAY)=0.
         TOP(NUMLAY)=6500.
         RETURN
   C     JUMP IF LOWEST BASE PRESENT.
      40 IF (HLOW .GT. 0.0) GO TO 70
   C     JUMP IF LOW CLOUD TYPE PRESENT.
         IF (LOWT .GT. 0) GO TO 60
   C     JUMP IF LOW CLOUD TYPE MISSING.
```

CFAS SUBPROGRAM ELEMENT SYNOP

```
      IF (LOWT .LT. 0) GO TO 50
C     CODE MIDDLE CLOUD
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=2
      COVER(NUMLAY)=CLOW
      BASE(NUMLAY)=BASMID
C     CODE CLEAR LAYER TO BASE.
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=6
      COVER(NUMLAY)=0.
      BASE(NUMLAY)=0.
      TOP(NUMLAY)=BASMID
      RETURN
C     CODE LOW CLOUD
   50 NUMLAY=NUMLAY+1
      KIND(NUMLAY)=1
      COVER(NUMLAY)=CLOW
      BASE(NUMLAY)=BASLOW
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=6
      COVER(NUMLAY)=0.
      BASE(NUMLAY)=0.
      TOP(NUMLAY)=BASLOW
      RETURN
C     CODE LOW CLOUD
   60 NUMLAY=NUMLAY+1
      KIND(NUMLAY)=1
      COVER(NUMLAY)=CLOW
      BASE(NUMLAY)=BASLOW
C     CODE CLEAR LAYER TO BASE
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=6
      COVER(NUMLAY)=0.
      BASE(NUMLAY)=0.
      TOP(NUMLAY)=BASLOW
      RETURN
C     JUMP IF NO LOW CLOUD.
   70 IF (HLOW .GT. 6500.0) GO TO 90
C     CODE LOW CLOUD
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=1
      COVER(NUMLAY)=CLOW
      BASE(NUMLAY)=AMAX1(HLOW,BASLOW)
```

CFAS SUBPROGRAM ELEMENT SYNOP

```
C     CODE LOWEST BASE IF OBSCURATION.

      IF (CLOW .LE. 0.99) GO TO 80
      IF (HLOW .LE. 0.0) GO TO 80
      BASE(NUMLAY)=HLOW

C     CODE CLEAR LAYER TO BASE

   80 NUMLAY=NUMLAY+1
      KIND(NUMLAY)=6
      COVER(NUMLAY)=0.
      BASE(NUMLAY)=0.
      TOP(NUMLAY)=BASE(NUMLAY-1)
      RETURN

C     CODE MIDDLE CLOUD

   90 NUMLAY=NUMLAY+1
      KIND(NUMLAY)=2
      COVER(NUMLAY)=CLOW
      BASE(NUMLAY)=HLOW

C     CODE CLEAR LAYER TO BASE

      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=6
      COVER(NUMLAY)=0.
      BASE(NUMLAY)=0.
      TOP(NUMLAY)=BASE(NUMLAY-1)
      RETURN

C     CODE ALL CLEAR.

  100 NUMLAY=NUMLAY+1
      KIND(NUMLAY)=6
      COVER(NUMLAY)=0.
      BASE(NUMLAY)=0.
      TOP(NUMLAY)=TOPCLR
      RETURN

C     JUMP IF TOTAL CLOUD COVER IS OVERCAST

  110 IF (CTOT .GT. 0.99) GO TO 480

C     JUMP IF LOWEST CLOUD COVER NOT MISSING AND NOT ZERO

      IF (CLOW .GT. 0.05) GO TO 480

C     CODE ALL CLEAR.

      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=6
      COVER(NUMLAY)=0.
      BASE(NUMLAY)=0.
      TOP(NUMLAY)=TOPCLR

C     JUMP IF LOWEST CLOUD COVER MISSING.

      IF (CLOW .LT. -1.0) GO TO 120
```

I-74

CFAS SUBPROGRAM ELEMENT SYNOP

```
      C     ASSUME NO LOW CLOUDS
            GO TO 240
      C     JUMP IF LOWEST BASE PRESENT.
        120 IF (HLOW .GT. 0.0) GO TO 320
      C     JUMP IF LOW CLOUD TYPE MISSING OR ZERO.
            IF (LOWT .LE. 0) GO TO 130
      C     CODE LOW CLOUD DEFINITELY PRESENT.
            G1=1.
            GOTO140
      C     JUMP IF LOW CLOUD TYPE ZERO.
        130 IF (LOWT .EQ. 0) GO TO 240
      C     CODE LOW CLOUD MIGHT BE PRESENT
            G1=0.5
      C     JUMP IF MIDDLE CLOUD TYPE MISSING OR EQUAL ZERO.
        140 IF (MIDT .LE. 0) GO TO 150
      C     CODE MIDDLE CLOUD DEFINITELY PRESENT.
            G2=1.
            GOTO160
      C     JUMP IF MIDDLE CLOUD TYPE ZERO.
        150 IF (MIDT .EQ. 0) GO TO 200
      C     CODE MIDDLE CLOUD MIGHT BE PRESENT.
            G2=0.5
      C     JUMP IF HIGH CLOUD TYPE MISSING OR ZERO.
        160 IF (NHIT .LE. 0) GO TO 170
      C     CODE HIGH CLOUD DEFINITELY PRESENT.
            G3=1.
            GOTO180
      C     JUMP IF TOTAL CLOUD COVER OVERCAST OR HIGH CLOUD ZERO
        170 IF (CTOT .GT. 0.98) GO TO 190
            IF (NHIT .EQ. 0) GO TO 190
      C     CODE HIGH CLOUD MIGHT BE PRESENT
            G3=0.5
      C     DETERMINE THREE RANDOM LAYERS.
```

```
  180 CALL CASE1(G1,G2,G3,CTOT,CLD1,CLD2,CLD3)

C     CODE LOW , MIDDLE, AND HIGH CLOUDS.

      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=1
      COVER(NUMLAY)=CLD1
      BASE(NUMLAY)=BASLOW
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=2
      COVER(NUMLAY)=CLD2
      BASE(NUMLAY)=BASMID
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=3
      COVER(NUMLAY)=CLD3
      BASE(NUMLAY)=BASHI
      RETURN

C     DETERMINE TWO RANDOM LAYERS.

  190 CALL CASE2(G1,G2,CTOT,CLD1,CLD2)

C     CODE LOW AND MIDDLE CLOUDS.

      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=1
      COVER(NUMLAY)=CLD1
      BASE(NUMLAY)=BASLOW
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=2
      COVER(NUMLAY)=CLD2
      BASE(NUMLAY)=BASMID
      RETURN

C     JUMP IF HIGH CLOUD TYPE MISSING OR ZERO.

  200 IF (NHIT .LE. 0) GO TO 210

C     CODE HIGH CLOUD DEFINITELY PRESENT.

      G3=1.
      GOTO220

C     JUMP IF HIGH CLOUD TYPE ZERO.

  210 IF (NHIT .EQ. 0) GO TO 230

C     CODE HIGH CLOUD MIGHT BE PRESENT.

      G3=0.5

C     DETERMINE TWO RANDOM LAYERS.

  220 CALL CASE2 (G1, G3, CTOT, CLD1, CLD3)

C     CODE LOW AND HIGH CLOUDS PRESENT.

      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=1
```

CFAS SUBPROGRAM ELEMENT SYNOP

```
            COVER(NUMLAY)=CLD1
            BASE(NUMLAY)=BASLOW
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=3
            COVER(NUMLAY)=CLD3
            BASE(NUMLAY)=BASHI
            RETURN

      C     CODE LOW CLOUD PRESENT.

        230 NUMLAY=NUMLAY+1
            KIND(NUMLAY)=1
            COVER(NUMLAY)=CTOT
            BASE(NUMLAY)=BASLOW
            RETURN

      C     JUMP IF MIDDLE CLOUD TYPE MISSING OR ZERO.

        240 IF (MIDT .LE. 0) GO TO 250

      C     CODE MIDDLE CLOUD DEFINITELY PRESENT.

            G2=1.
            GOTO260

      C     JUMP IF MIDDLE CLOUD TYPE ZERO.

        250 IF (MIDT .EQ. 0) GO TO 300

      C     CODE MIDDLE CLOUD MIGHT BE PRESENT.

            G2=0.5

      C     JUMP IF HIGH CLOUD TYPE MISSING OR ZERO.

        260 IF (NHIT .LE. 0) GO TO 270

      C     CODE HIGH CLOUD DEFINITELY PRESENT.

            G3=1.
            GOTO280

      C     BUILD MIDDLE LAYER ONLY IF TOTAL IS OVERCAST OR HIGH TYPE IS ZERO.

        270 IF (CTOT .GT. 0.98) GO TO 290
            IF (NHIT .EQ. 0) GO TO 290

      C     CODE HIGH CLOUD MIGHT BE PRESENT.

            G3=0.5

      C     DETERMINE TWO RANDOM CLOUD LAYERS.

        280 CALL CASE2(G2,G3,CTOT,CLD2,CLD3)

      C     CODE MIDDLE AND HIGH CLOUDS.

            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=2
```

CFAS SUBPROGRAM ELEMENT SYNOP

```
          COVER(NUMLAY)=CLD2
          BASE(NUMLAY)=BASMID
          NUMLAY=NUMLAY+1
          KIND(NUMLAY)=3
          COVER(NUMLAY)=CLD3
          BASE(NUMLAY)=BASHI
          RETURN
C         CODE MIDDLE CLOUD
   290    NUMLAY=NUMLAY+1
          KIND(NUMLAY)=2
          COVER(NUMLAY)=CTOT
          BASE(NUMLAY)=BASMID
          RETURN
C         CODE HIGH CLOUD.
   300    NUMLAY=NUMLAY+1
          KIND(NUMLAY)=3
          COVER(NUMLAY)=CTOT
          BASE(NUMLAY)=BASHI
C    BUILD CLEAR TO TOP IF TOTAL CLOUD OVERCAST AND HIGH TYPE MSG OR ZERO.
          IF (CTOT .LE. 0.98) GO TO 310
          IF (NHIT .LE. 0) GO TO 100
   310    RETURN
C         JUMP IF NO LOW CLOUD
     320 IF (HLOW .GT. 6500.0) GO TO 430
C         CODE LOW CLOUD DEFINITELY PRESENT.
          G1=1.
C         JUMP IF MIDDLE CLOUD TYPE MISSING OR ZERO.
          IF (MIDT .LE. 0) GO TO 330
C         CODE MIDDLE CLOUD DEFINITELY PRESENT.
          G2=1.
          GOTO 340
C         JUMP IF MIDDLE CLOUD TYPE ZERO.
     330 IF (MIDT .EQ. 0) GO TO 380
C         CODE MIDDLE CLOUD TYPE MIGHT BE PRESENT.
          G2=0.5
C         JUMP IF HIGH CLOUD TYPE MISSING OR ZERO.
     340 IF (NHIT .LE. 0) GO TO 350
C         CODE HIGH CLOUD DEFINITELY PRESENT.
```

```
      G3=1.
      GOTO360

C     HIGH TYPE MISSING OR ZERO, CODE MIDDLE CLOUD IF TOTAL OVERCAST.

  350 IF (CTOT .GT. 0.98) GO TO 370
      IF (NHIT .EQ. 0) GO TO 370

C     CODE HIGH CLOUD MIGHT BE PRESENT.

      G3=0.5

C     DETERMINE THREE RANDOM CLOUD LAYERS.

  360 CALL CASE1(G1,G2,G3,CTOT,CLD1,CLD2,CLD3)

C     CODE LOW, MIDDLE, AND HIGH CLOUDS.

      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=1
      COVER(NUMLAY)=CLD1
      BASE(NUMLAY)=AMAX1(BASLOW,HLOW)
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=2
      COVER(NUMLAY)=CLD2
      BASE(NUMLAY)=BASMID
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=3
      COVER(NUMLAY)=CLD3
      BASE(NUMLAY)=BASHI
      RETURN

C     DETERMINE TWO RANDOM CLOUD LAYERS.

  370 CALL CASE2(G1,G2,CTOT,CLD1,CLD2)

C     CODE LOW AND MIDDLE CLOUDS.

      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=1
      COVER(NUMLAY)=CLD1
      BASE(NUMLAY)=AMAX1(BASLOW,HLOW)
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=2
      COVER(NUMLAY)=CLD2
      BASE(NUMLAY)=BASMID
      RETURN

C     JUMP IF HIGH CLOUD TYPE MISSING OR ZERO.

  380 IF (NHIT .LE. 0) GO TO 390

C     CODE HIGH CLOUD DEFINITELY PRESENT.

      G3=1.
      GO TO 400

C     JUMP IF HIGH CLOUD TYPE ZERO.
```

```
      390 IF (NHIT .EQ. 0) GO TO 420
      C     CODE HIGH CLOUD MIGHT BE PRESENT.
            G3=0.5
      C     DETERMINE TWO RANDOM CLOUD LAYERS.
      400 CALL CASE2(G1,G3,CTOT,CLD1,CLD3)
      C     CODE LOW AND HIGH CLOUDS.
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=1
            COVER(NUMLAY)=CLD1
            BASE(NUMLAY)=AMAX1(BASLOW,HLOW)
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=3
            COVER(NUMLAY)=CLD3
            BASE(NUMLAY)=BASHI
      C     CLEAR TO TOP IF TOTAL OVERCAST AND HIGH TYPE MISSING
            IF (CTOT .LE. 0.98) GO TO 410
            IF (NHIT .LT. 0) GO TO 100
      410 RETURN
      C     CODE LOW CLOUD
      420 NUMLAY=NUMLAY+1
            KIND(NUMLAY)=1
            COVER(NUMLAY)=CTOT
            BASE(NUMLAY)=AMAX1(BASLOW,HLOW)
            RETURN
      C     CODE MIDDLE CLOUD DEFINITELY PRESENT.
      430 G2=1.
      C     JUMP IF HIGH CLOUD TYPE MISSING OR ZERO.
            IF (NHIT .LE. 0) GO TO 440
      C     CODE HIGH CLOUD DEFINITELY PRESENT.
            G3=1.
            GOTO 450
      C     JUMP IF HIGH CLOUD TYPE ZERO.
      440 IF (NHIT .EQ. 0) GO TO 460
      C     CODE HIGH CLOUD MIGHT BE PRESENT.
            G3=0.5
      C     DETERMINE TWO RANDOM CLOUD LAYERS.
      450 CALL CASE2(G2,G3,CTOT,CLD2,CLD3)
```

```

      C      CODE MIDDLE AND HIGH CLOUDS.

             NUMLAY=NUMLAY+1
             KIND(NUMLAY)=2
             COVER(NUMLAY)=CLD2
             BASE(NUMLAY)=BASMID
             NUMLAY=NUMLAY+1
             KIND(NUMLAY)=3
             COVER(NUMLAY)=CLD3
             BASE(NUMLAY)=BASHI
             RETURN

      C      CODE MIDDLE CLOUD

         460 NUMLAY=NUMLAY+1
             KIND(NUMLAY)=2
             COVER(NUMLAY)=CTOT
             BASE(NUMLAY)=BASMID
             RETURN

      C      JUMP IF LOW CLOUD AMOUNT IS MISSING.

         470 IF (CLOW .LT. -1.0) GO TO 120

      C      NO LOW CLOUDS, TEST MIDDLE AND HIGH TYPES AND TOTAL CLOUD COVER.

             GO TO 240

      C      JUMP IF LOW CLOUD IS OVERCAST.

         480 IF (CLOW .GT. 0.99) GO TO 40

      C      JUMP IF TOTAL IS NOT OVERCAST.

             IF (CTOT .LT. 0.99) GO TO 490
             CTOT=0.99

      C      TOTAL IS OVERCAST, JUMP IF HIGH CLOUD TYPE GIVEN.

             IF (NHIT .GT. 0) GO TO 490

      C      HIGH CLOUD UNKNOWN, CLEAR ONLY TO BASMID.

             NUMLAY=NUMLAY+1
             KIND(NUMLAY)=6
             COVER(NUMLAY)=0.
             BASE(NUMLAY)=0.
             TOP(NUMLAY)=BASMID

      C      JUMP FOR LOW CLOUD AMOUNT MISSING OR ZERO.
             IF(CLOW -0.05) 470,500,500

      C      CODE ALL CLEAR.

         490 NUMLAY=NUMLAY+1
             KIND(NUMLAY)=6
             COVER(NUMLAY)=0.
             BASE(NUMLAY)=0.
```

```
                TOP(NUMLAY)=TOPCLR
C               JUMP FOR LOW CLOUD AMOUNT MISSING OR ZERO.
                IF (CLOW .LT. 0.05) GO TO 470
C               ASSURE LOWEST CLOUD COVER LESS THAN TOTAL CLOUD COVER REQUIRED.
C                    IN 'CASES'.
            500 CTEMP=CTOT-0.01
                CLOW=AMIN1(CLOW,CTEMP)
C               JUMP IF LOW CLOUD TYPE PRESENT.
                IF (LOWT .GT. 0) GO TO 570
C               JUMP IF LOW CLOUD TYPE MISSING AND LOWEST BASE LT 6500 FT.
                IF (LOWT .LT. 0) GO TO 510
                IF (HLOW .LT. 6500.0) GO TO 560
C               JUMP IF HIGH CLOUD TYPE PRESENT.
            510 IF (NHIT .GT. 0) GO TO 540
C               JUMP IF TOTAL NOT OVERCAST AND HIGH TYPE IS MISSING.
                IF (NHIT .LT. 0) GO TO 520
                IF (CTOT .LT. 0.98) GO TO 540
C               CODE MIDDLE CLOUD
            520 NUMLAY=NUMLAY+1
                KIND(NUMLAY)=2
                COVER(NUMLAY)=CTOT
                BASE(NUMLAY)=BASMID
                IF (HLOW .LE. 6500.0) GO TO 530
                BASE(NUMLAY)=HLOW
            530 RETURN
C               DETERMINE TWO RANDOM CLOUD LAYERS.
            540 CALL CASE6(CLOW,CTOT,CLD2,CLD3)
C               CODE MIDDLE AND HIGH CLOUDS.
                NUMLAY=NUMLAY+1
                KIND(NUMLAY)=2
                COVER(NUMLAY)=CLD2
                BASE(NUMLAY)=BASMID
                IF (HLOW .LE. 6500.0) GO TO 550
                BASE(NUMLAY)=HLOW
            550 NUMLAY=NUMLAY+1
                KIND(NUMLAY)=3
                COVER(NUMLAY)=CLD3
                BASE(NUMLAY)=BASHI
                RETURN
C               CODE LOW CLOUDS MIGHT BE PRESENT.
```

CFAS SUBPROGRAM ELEMENT SYNOP

```
  560 G1=0.5
      GOTO600
C     JUMP IF CB.
  570 IF (LOWT .EQ. 3) GO TO 580
      IF (LOWT .EQ. 9) GO TO 580
C     JUMP IF TCU
      IF (LOWT .EQ. 2) GO TO 590
C     CODE LOW CLOUD DEFINITELY PRESENT.
      G1=1.
      GOTO600
C     CODE CB PRESENT AND ASSURE THUNDERSTORM IN WEATHER.
  580 NCB=3
      NWEA(7)=MAX0(NWEA(7),90)
      GO TO 595
C     CODE TCU PRESENT AND ASSURE SHOWER IN WEATHER.
  590 NCB=2
      NWEA(7)=MAX0(NWEA(7),80)
  595 IF(MSPW .LT. NWEA(7)) MSPW=NWEA(7)
C     JUMP IF MIDDLE CLOUD TYPE PRESENT.
  600 IF (MIDT .GT. 0) GO TO 710
C     JUMP IF MIDDLE CLOUD TYPE MISSING.
      IF (MIDT .LT. 0) GO TO 700
C     JUMP IF HIGH CLOUD TYPE PRESENT.
      IF (NHIT .GT. 0) GO TO 630
C     JUMP IF HIGH CLOUD TYPE MISSING.
      IF (NHIT .LT. 0) GO TO 630
C     CODE LOW AND POSSIBLY MIDDLE AND HIGH CLOUDS.
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=1
      COVER(NUMLAY)=CTOT
      BASE(NUMLAY)=AMAX1(BASLOW,HLOW)
C     RETURN IF NO CB OR TCU
      IF (NCB .GE. 2) GO TO 610
      RETURN
  610 NUMLAY=NUMLAY+1
      KIND(NUMLAY)=2
      COVER(NUMLAY)=CTOT*REDUCE
      BASE(NUMLAY)=BASMID
```

CFAS SUBPROGRAM ELEMENT SYNOP

```
C      RETURN IF NO CB

       IF (NCB .GE. 3) GO TO 620
       RETURN
   620 NUMLAY=NUMLAY+1
       KIND(NUMLAY)=3
       COVER(NUMLAY)=CTOT*REDUCE**2
       BASE(NUMLAY)=BASHI
       RETURN

C      JUMP ON NEITHER OR TCU OR CB.

   630 GO TO (640, 660, 680), NCB

C      DETERMINE TWO RANDOM LAYERS.

   640 CALL CASE6(CLOW,CTOT,CLD1,CLD3)

C      CODE LOW AND HIGH CLOUDS.

       NUMLAY=NUMLAY+1
       KIND(NUMLAY)=1
       COVER(NUMLAY)=CLD1
       BASE(NUMLAY)=AMAX1(BASLOW,HLOW)
       NUMLAY=NUMLAY+1
       KIND(NUMLAY)=3
       COVER(NUMLAY)=CLD3
       BASE(NUMLAY)=BASHI

C      BUILD CLEAR LAYER TO TOP IF TOTAL IS OVERCAST AND
C      HIGH TYPE NOT GIVEN

       IF (CTOT .LE. 0.98) GO TO 650
       IF (NHIT .LE. 0) GO TO 100
   650 RETURN

C      CODE LOW CLOUD.

   660 NUMLAY=NUMLAY+1
       KIND(NUMLAY)=1
       COVER(NUMLAY)=CLOW
       BASE(NUMLAY)=AMAX1(BASLOW,HLOW)

C      DETERMINE TWO RANDOM CLOUD LAYERS.
       CALL CASE6(CLOW,CTOT,CLD2,CLD3)

C      CODE MIDDLE AND HIGH CLOUDS.

       NUMLAY=NUMLAY+1
       KIND(NUMLAY)=2
       COVER(NUMLAY)=CLD2*REDUCE
       BASE(NUMLAY)=BASMID
       NUMLAY=NUMLAY+1
       KIND(NUMLAY)=3
       COVER(NUMLAY)=CLD3
       BASE(NUMLAY)=BASHI

C  BUILD CLEAR LAYER TO TOP IF TOTAL IS OVERCAST AND HIGH TYPE NOT GIVEN
```

```
            IF (CTOT .LE. 0.98) GO TO 670
            IF (NHIT .LE. 0) GO TO 100
     670    RETURN

      C     CODE LOW, MIDDLE AND HIGH CLOUDS.

     680    NUMLAY=NUMLAY+1
            KIND(NUMLAY)=1
            COVER(NUMLAY)=CLOW
            BASE(NUMLAY)=AMAX1(BASLOW,HLOW)

      C     DETERMINE TWO RANDOM CLOUD LAYERS.

            CALL CASE6(CLOW,CTOT,CLD2,CLD3)
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=2
            COVER(NUMLAY)=CLD2*REDUCE
            BASE(NUMLAY)=BASMID
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=3
            COVER(NUMLAY)=CLD2*REDUCE**2+(1.-CLD2*REDUCE**2)*CLD3
            BASE(NUMLAY)=BASHI

      C  BUILD CLEAR LAYER TO TOP IF TOTAL IS OVERCAST AND HIGH TYPE NOT GIVEN

            IF (CTOT .LE. 0.98) GO TO 690
            IF (NHIT .LE. 0) GO TO 100
     690    RETURN

      C     CODE MIDDLE CLOUD TYPE MIGHT BE PRESENT.

     700    G2=0.5
            GOTO720

      C     CODE MIDDLE CLOUD DEFINITELY PRESENT.

     710    G2=1.

      C     JUMP IF HIGH CLOUD TYPE PRESENT.

     720    IF (NHIT .GT. 0) GO TO 800

      C     JUMP ON MISSING HIGH TYPE ONLY IF TOTAL IS NOT OVERCAST.

            IF (NHIT .GE. 0) GO TO 730
            IF (CTOT .LT. 0.98) GO TO 770

      C     JUMP ON EITHER TCU OR CB.

     730    GO TO(740,750,760),NCB

      C     DETERMINE TWO RANDOM CLOUD LAYERS.

     740    CALL CASE6(CLOW,CTOT,CLD1,CLD2)

      C     CODE LOW AND MIDDLE CLOUDS.

            NUMLAY=NUMLAY+1
```

```
            KIND(NUMLAY)=1
            COVER(NUMLAY)=CLD1
            BASE(NUMLAY)=AMAX1(BASLOW,HLOW)
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=2
            COVER(NUMLAY)=CLD2
            BASE(NUMLAY)=BASMID
            RETURN

C           CODE LOW AND MIDDLE CLOUDS.

C           DETERMINE TWO RANDOM CLOUD LAYERS.

        750 CALL CASE6 (CLOW, CTOT, CLD1, CLD2)
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=1
            COVER(NUMLAY)=CLD1
            BASE(NUMLAY)=AMAX1(BASLOW,HLOW)
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=2
            COVER(NUMLAY)=CLD1*REDUCE+(1.-CLD1*REDUCE)*CLD2
            BASE(NUMLAY)=BASMID
            RETURN

C           CODE LOW MIDDLE AND HIGH CLOUDS.

C           DETERMINE TWO RANDOM CLOUD LAYERS.

        760 CALL CASE6(CLOW,CTOT,CLD1,CLD2)
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=1
            COVER(NUMLAY)=CLD1
            BASE(NUMLAY)=AMAX1(BASLOW,HLOW)
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=2
            COVER(NUMLAY)=CLD1*REDUCE+(1.-CLD1*REDUCE)*CLD2
            BASE(NUMLAY)=BASMID
            NUMLAY=NUMLAY+1
            KIND(NUMLAY)=3
            COVER(NUMLAY)=CLD1*REDUCE**2
            BASE(NUMLAY)=BASHI
            RETURN

C           CODE HIGH CLOUD MIGHT BE PRESENT.

        770 G3=0.5
            GO TO 810

C           CODE HIGH CLOUD DEFINITELY PRESENT.

        800 G3=1.

C           JUMP ON EITHER TCU OR CB.

        810 GO TO (820, 840, 850), NCB

C           DETERMINE THREE RANDOM CLOUD LAYERS.

        820 CALL CASE5(G2,G3,CLOW,CTOT,CLD1,CLD2,CLD3)
```

```
C     CODE LOW, MIDDLE, AND HIGH CLOUDS.

  830 NUMLAY=NUMLAY+1
      KIND(NUMLAY)=1
      COVER(NUMLAY)=CLD1
      BASE(NUMLAY)=AMAX1(BASLOW,HLOW)
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=2
      COVER(NUMLAY)=CLD2
      BASE(NUMLAY)=BASMID
      NUMLAY=NUMLAY+1
      KIND(NUMLAY)=3
      COVER(NUMLAY)=CLD3
      BASE(NUMLAY)=BASHI
      RETURN

C     DETERMINE THREE RANDOM CLOUD LAYERS WITH TCU.

  840 CALL CASE3(G2,G3,CLOW,CTOT,CLD1,CLD2,CLD3,REDUCE)
      GO TO 830

C     DETERMINE THREE RANDOM CLOUD LAYERS WITH CB.

  850 CALL CASE4(G2,G3,CLOW,CTOT,CLD1,CLD2,CLD3,REDUCE)
      GO TO 830
      END
```

CFAS SUBPROGRAM ELEMENT TOPS

CLOUD-FOG*CFAS.TOPS

```
      SUBROUTINE TOPS(TERHT,NWEA,DLAT)
C
C     ROUTINE TO DETERMINE CLOUD TOPS GIVEN CLOUD BASES, CLOUD COVER,
C          AND WEATHER.
C
C     TERHT = TERRAIN HEIGHT IN FEET
C     NWEA = WEATHER IN AREA (WMO CODE 4677)
C     WEAHIT = EXPECTED HIEGHTS OF CLCUD TOPS IN 100 S OF FEET DUE TO
C          WEATHER
C     KCURW = WEATHER FACTORS FOR WX 50-99
C     KPWEA = WEATHER FACTORS WX 10-29
C     THICKC = THICKNESS OF CLOUD IN FEET AT MSL
C     STHICK = SLOPE OF CLOUD THICKNESS WITH RESPECT TO BASE OF CLOUD
C          ABOVE MSL
C     CLOTOP = MAXIMUM HEIGHT OF CLOUD TOP IN FEET
C     SAMT = CONVERSION FACTOR FOR CLOUD COVER TO CLOUD THICKNESS
C        FACTOR
C     DLAT = LATITUDE
C
C DERIVED LAYERED CLOUD INFORMATION
C
C  NUMLAY = NUMBER OF LAYERS GENERATED
C  KIND = KIND OF CLOUD LAYER
C                  1 = LOW
C                  2 = MIDDLE
C                  3 = HIGH
C                  4 = FOG
C                  5 = LOWEST CLOUD
C                  6 = CLEAR LAYER
C  ITHIN = THIN LAYER DESIGNATOR
C                  MISSING = NOT THIN
C                  1 = THIN
C  COVER = CLOUD COVER IN LAYER (0.0 - 1.0)
C  BASE = HEIGHT OF THE BASE OF LAYER, FEET.
C  TOP = HEIGHT OF TOP OF CLOUD LAYER, FEET.
C
      DIMENSION NWEA(7),WEAHIT(3),KCURW(5),KPWEA(20),THICKD(7),STHICK(7)

      COMMON/CLOUDS/NUMLAY,KIND(10),ITHIN(10),COVER(10),BASE(10),TOP(10)

      DATA
     *(WEAHIT(K),K=1,2)/9000.,14000./
     *SAMT/1.5/
     *KCURW/1,2,2,2,3/
     *KPWEA /1, 0,0,0,0, 1, 2,2,2, 3, 1,1,1, 2, 1, 2,2,2, 0, 2/
     *THICKD/0.,1287.,2843.,4323.,5864.,7636.,9843./
     *STHICK/0.,0.13108,0.25523,0.41947,0.62827,0.87444,1.1191/

C     CALCULATE MAXIMUM CLOUD TOP

      CLDTOP=40000.-(10000./90.)*ABS(DLAT)

C     CALCULATE WEATHER FACTOR 3 HEIGHT.

      WEAHIT(3)=35000.-(13000./90.)*ABS(DLAT)

C     RETURN IF NO CLOUD LAYERS

```

```
            IF(NUMLAY .GT. 0) GO TO 10
            RETURN

      C     SET WEATHER FACTOR TO ZERO

         10 KWEA =0

      C     STEP THRU WEATHER

            DO 30 NUMWEA=1,7

      C     JUMP IF WEATHER LT 50

            IF(NWEA(NUMWEA) .LT. 50) GO TO 20

      C     DETERMINE WEATHER FACTOR.

            INDEX=NWEA(NUMWEA)/10-4
            KWEA=MAX0(KWEA,KCURW(INDEX))
            GO TO 30

      C     JUMP IF WEATHER GT 29 OR LT 10

         20 IF(NWEA(NUMWEA) .GT. 29) GO TO 30
            IF(NWEA(NUMWEA) .LT. 10) GO TO 30

      C     DETERMINE WEATHER FACTOR

            INDEX=NWEA(NUMWEA)-9
            KWEA=MAX0(KWEA,KPWEA(INDEX))
         30 CONTINUE

      C     STEP THRU CLOUD LAYERS

            DO 60 LAY=1,NUMLAY

      C     JUMP ON KIND OF CLOUD

            ISWIT=KIND(LAY)
            GO TO (40, 40, 40, 40, 50, 60), ISWIT

      C     JUMP IF NO SIGNIFICANT WEATHER PRESENT.

         40 IF(KWEA .EQ. 0) GO TO 50

      C     DETERMINE CLOUD AMOUNT THICKNESS FACTOR

            FACAMT=SAMT*COVER(LAY)

      C     DETERMINE TOTAL CLOUD THICKNESS FACTORS

            FACT=FACAMT+KWEA
            KFACT=FACT+1.
            DFACT=FACT+1.-KFACT

      C     CALCULATE CLOUD TOP IN FEET ABOVE TERRAIN

            TOP(LAY)=BASE(LAY)+THICKD(KFACT)+STHICK(KFACT)*(BASE(LAY)+TERHT)+
           *DFACT*(THICKD(KFACT+1)+STHICK(KFACT+1)*(BASE(LAY)+TERHT))
```

CFAS SUBPROGRAM ELEMENT TOPS

```
      C     LIMIT CLOUD TOP

            TOP(LAY)=AMIN1(TOP(LAY),CLDTOP)

      C     JUMP IF FOG LAYER

            IF(ISWIT .EQ. 4) GO TO 60

      C     ACCOUNT FOR WEATHER IN ITS OWN RIGHT.

            TOP(LAY)=AMAX1(TOP(LAY),WEAHIT(KWEA))
            GO TO 60

      C     JUMP IF FOG LAYER.

         50 IF(ISWIT .EQ. 4) GO TO 60

      C     DETERMINE CLOUD AMOUNT THICKNESS FACTOR

            FACAMT=SAMT*COVER(LAY)

      C     CALCULATE TOTAL CLOUD THICKNESS FACTORS.

            KFACT=FACAMT+1.
            DFACT=FACAMT+1.-KFACT

      C     CALCULATE CLOUD TOP IN FEET ABOVE TERRAIN

            TOP(LAY)=BASE(LAY)+THICKO(KFACT)+STHICK(KFACT)*(BASE(LAY)+TERHT)+
           *DFACT*(THICKO(KFACT+1)+STHICK(KFACT+1)*(BASE(LAY)+TERHT))

      C     LIMIT CLOUD TOP

            TOP(LAY)=AMIN1(TOP(LAY),CLDTOP)
         60 CONTINUE
            RETURN
            END
```

CFAS SUBPROGRAM ELEMENT UADINT

```
CLOUD-FOG*CFAS.UADINT
          SUBROUTINE UADINT

      C   ROUTINE TO INTERPRET UPPER AIR OBS/REP IN TERMS OF CFDB PARAMETERS

      C   SOURCES OF INPUT DATA ARE UPPER AIR SOUNDINGS (RAOBS) OF PRESSURE,
      C       TEMPERATURE AND DEWPOINT DEPRESSION.

      C   INPUT DATA

      C   IX = X DISTANCE OF RAOB SITE FROM IXREF, HECTOMETERS.
      C   IY = Y DISTANCE OF RAOB SITE FROM IYREF, HECTOMETERS.
      C   IH = STATION ELEVATION ABOVE MEAN SEA LEVEL, METERS.
      C   ITIME = TIME OF RAOB, (0-1440)
      C   ITYPE = 4, (-4 IF A SPECIAL RAOB)
      C   IZ(I)  = ALTITUDE OF RAOB REPORTING LEVEL, METERS
      C   IP(I)  = PRESSURE OF RAOB REPORTING LEVELS, MILLIBARS*10
      C   IT(I)  = TEMPERATURE OF RAOB REPORTING LEVEL, (DEG. K.)*10
      C   IDD(I) = DEWPOINT DEPRESSION OF RAOB REPORTING LEVEL, (DEG. C)*10
      C   NRRL   = NUMBER OF RAOB REPORTING LEVELS

      C   CLOUD/FOG DATA BASE PARAMETERS

      C   IVALU = INFORMATION VALUE OF THE RAOB (1-10)
      C        0 = NO CFDB PARAMETERS OBTAINABLE FROM THE RAOB.
      C       10 = NO MISSING OR INCONSISTENT DATA IN THE RAOB.
      C     0-10 = SOME MISSING OR INCONSISTENT DATA IN THE RAOB.
      C   MINBAS = HEIGHT OF THE BASE OF THE LOWEST CLOUD (AGL), DEKAMETERS.
      C   MAXTOP = HEIGHT OF THE TOP OF THE HIGHEST CLOUD (AGL), DEKAMETERS.
      C   LCOV(9) = PERCENT CLOUD COVER IN THE CFDB LAYERS.

          COMMON /OBSREP/ IX,IY,IH,ITIME,IOBC,ITYPE,IVALU,NU(3),MINBAS,
         *MAXTOP,NLV,LCOV(9),IZ(30),IP(30),IT(30),IDD(30),NRRL

          DIMENSION HMP(9),HLEV(10),PMP(9),TMP(9),DMP(9)

          DATA HLEV/0.,150.,300.,600.,1000.,2000.,3500.,5000.,6500.,10000./

          DATA MISS/-32768/

          DO 1 LAY=1,9
        1 LCOV(LAY)=MISS

          ITC=0
          DO 20 LEV=2,NRRL

      C   IGNORE REPORT IF INCORRECT PRESSURE CONVENTION

          IF(IP(LEV) .LT. IABS(IP(LEV-1)))  GO TO 5
          VALU=0.
          GO TO 200

      C   IGNORE LEVEL IF PRESSURE IS MISSING

        5 IF(IP(LEV) .NE. MISS) GO TO 10
          NRRL=NRRL-1
          DO 7 J=LEV,NRRL
          JA=J+1
          IZ(J)=IZ(JA)
```

I-91

CFAS SUBPROGRAM ELEMENT UADINT

```
          IP(J)=IP(JA)
          IT(J)=IT(JA)
        7 IDD(J)=IDD(JA)
       10 CONTINUE
          IF(IT(LEV) .GT. 0) ITC=ITC+1
       20 CONTINUE

C         IGNORE REPORT IF TEMPERATURE NOT SPECIFIED AT TWO LEVELS

          IF(ITC .GE. 2) GO TO 30
          VALU=0.
          GO TO 200

C         INSURE A TEMPERATURE SPECIFICATION AT THE HIGHEST LEVEL.

       30 IF(IT(NRRL) .NE. MISS) GO TO 40
          NRRL=NRRL-1

C         IGNORE REPORT IF RAOB SITE ELEVATION IS MISSING

       40 IF(IH .NE. MISS) GO TO 50
          VALU=0.
          GO TO 200
       50 IZ(1)=IH

C         CALCULATE HEIGHT IN METERS ABOVE MEAN SEA LEVEL OF THE MIDPOINTS
C              OF THE CFDB LAYERS

          TRH=IH
          DO 60 LAY=1,9
       60 HMP(LAY)=((HLEV(LAY)+HLEV(LAY+1))*.1524)-TRH

C         DETERMINE TEMPERATURE AND DEW POINT SPREAD FOR THE CFDB LAYERS.

          CALL RAOB(HMP,PMP,TMP,DMP,VAL)

C         DETERMINE CLOUD COVER FOR CFDB LAYERS

          CALL DEFCLD(PMP,TMP,DMP,LCOV)

C         DETERMINE LOWEST AND HIGHEST LAYERS WITH CLOUDS

          MINBAS=10
          MAXTOP=0
          DO 70 LAY=1,9
          LAYR=10-LAY
          IF(LCOV(LAYR) .NE. MISS .AND. LCOV(LAYR) .GT. 0) MINBAS=MIN0(MINBA
         *S,LAYR)
          IF(LCOV(LAY) .NE. MISS .AND. LCOV(LAY) .GT. 0) MAXTOP=MAX0(MAXTOP,
         *LAY)
       70 CONTINUE

C         CALCULATE BASE AND TOPS OF CLOUD LAYERS IN DEKAMETERS

          IF(MAXTOP .LE. 0) MAXTOP=MISS
          IF(MINBAS .GE. 10) MINBAS=MISS
          IF(MINBAS .EQ. MISS) GO TO 80
          MINBAS=HLEV(MINBAS)*.03048
       80 IF(MAXTOP .EQ. MISS) GO TO 90
```

CFAS SUBPROGRAM ELEMENT UADINT

```
          MAXTOP=HLEV(MAXTOP+1)*.03048

   C      CALCULATE VALUE OF OBS/REP

       90 MLAY=0
          DO 100 LAY=1,9
          IF(LCOV(LAY) .EQ. MISS) MLAY=MLAY-1
      100 CONTINUE
          IF (MLAY .NE. 0) GO TO 110
          VALU=0.
          GO TO 200
      110 IF(MLAY .GE. 1) VALU=6.
          IF(MLAY .GE. 3) VALU=8.
          IF(MLAY .GE. 6) VALU=10.
          VALU=((5.*VALU) + VAL)/6.
      200 IVALU=VALU
          RETURN
          END
```

CFAS SUBPROGRAM ELEMENT UTM

CLOUD-FOG*CFAS.UTM

```
                SUBROUTINE  UTM(LON,LAT,EAST,NORTH,CMRD)
          C  CONVERTS LONGITUDE AND LATITUDE TO UTM EASTING AND NORTHING.
          C  CMRD IS LONGITUDE OF CENTRAL MERIDIAN.
                REAL LAT
                REAL LON
                REAL NORTH
                A = 63.782064
                ARED = 63.350345
                E = .0068147849
                Q = .017453292  * LAT
                P = 3600. * (CMRD - LON)
                C = COS(Q)
                S = SIN(Q)
                T = S/C
                S2 = 2. * S * C
                D = 1. - (2. * S * S)
                S4 = 2. * S2 * D
                RHO = A / SQRT(1. - (6.7686580E-03*S*S))
                D = Q + (.005076492 * (Q - (.5 * S2))) +
               $       (4.29513E-05 * ((1.5 * Q) - S2 + (S4 / 8.)))
                XN1 = ARED * D
                D = C * S * 1.1752215E-11*P*P
                D = D + ((C**3) * S * 2.3015189E-23*(P**4)*
               $       (5. - (T*T) + (9. * ((E*C)**2)) + (4. * ((E*C)**4))))
                NORTH = .9996 * (XN1 + (D * RHO))
                D = C * 4.8481368E-06*P
                D = D + ((C**3) * (1. - (T*T) + ((E*C)**2)) *
               $       1.8992115E-17*(P**3))
                EAST = (RHO * D * .9996) + 5.
                RETURN
                END
```

CFAS DATA ELEMENT OBSREP

CLOUD-FOG*CFAS.OBSREP

1	1288	1126	0	105	1	10000	2
2	21	-32768	-32768				
3	61	4	-32768	-32768	-32768	-32768	-32768
4	3	-32768	15	-32768			
5	8	-32768	60	-32768			
6	1943	23	0	117	1	10000	4
7	42	-32768	-32768				
8	180	60	51	-32768	-32768	-32768	-32768
9	3	-32768	55	-32768			
10	3	-32768	73	-32768			
11	6	-32768	110	1			
12	8	-32768	230	-32768			
13	576	1197	0	19	1	900	4
14	42	-32768	-32768				
15	180	60	51	43	-32768	-32768	-32768
16	3	-32768	55	-32768			
17	3	-32768	73	-32768			
18	6	-32768	110	1			
19	8	-32768	230	-32768			
20	1773	1594	0	1423	1	10000	4
21	22	1	1				
22	1	-32768	-32768	-32768	-32768	-32768	-32768
23	3	-32768	73	-32768			
24	8	-32768	34	-32768			
25	6	-32768	93	-32768			
26	8	-32768	115	-32768			
27	959	1891	0	1	1	10000	4
28	-32768	-32768	-32768				
29	163	82	-32768	-32768	-32768	-32768	-32768
30	3	-32768	43	-32768			
31	3	-32768	-10	0			
32	6	-32768	71	1			
33	8	-32768	161	103			
34	1283	1726	0	90	1	10000	1
35	13	1	-32768				
36	3	-32768	-32768	-32768	-32768	-32768	-32768
37	8	-32768	-12	-32768			
38	1844	1898	0	101	1	10000	2
39	-32768	-32768	-32768				
40	90	-32768	-32768	-32768	-32768	-32768	-32768
41	5	-32768	46	-32768			
42	7	-32768	-73	-32768			
43	406	1187	0	133	1	10000	2
44	-32768	-32768	-32768				
45	90	42	-32768	-32768	-32768	-32768	-32768
46	5	-32768	46	-32768			
47	7	-32768	-73	-32768			
48	759	785	0	1407	1	5000	2
49	-32768	-32768	-32768				
50	90	42	44	-32768	-32768	-32768	-32768
51	5	-32768	46	-32768			
52	7	-32768	-73	-32768			
53	681	675	0	130	1	1000	2
54	-32768	-32768	-32768				
55	90	42	44	46	-32768	-32768	-32768
56	5	-32768	46	-32768			
57	7	-32768	-73	-32768			
58	1069	993	0	39	2	2000	4

CFAS DATA ELEMENT OBSREP

59	95	-32768	-32768	-32768	-32768	-32768	-32768
60	2	8	18	-32768			
61	4	5	54	1			
62	6	2	75	10			
63	3	0	103	0			
64	1003	1233	0	60	-2	3000	4
65	82	43	-32768	-32768	-32768	-32768	-32768
66	4	6	18	-32768			
67	3	4	-20	-32768			
68	1	3	62	-32768			
69	8	1	87	-32768			
70	1427	1540	0	52	2	1500	4
71	82	43	44	-32768	-32768	-32768	-32768
72	4	6	18	-32768			
73	3	4	-20	-32768			
74	1	3	62	-32768			
75	8	1	87	-32768			
76	355	534	0	1413	-2	750	4
77	82	43	44	46	-32768	-32768	-32768
78	4	6	18	-32768			
79	3	4	-20	-32768			
80	1	3	62	-32768			
81	8	1	87	-32768			
82	793	1115	0	56	-2	200	4
83	82	43	44	46	45	-32768	-32768
84	4	6	18	-32768			
85	3	4	-20	-32768			
86	1	3	62	-32768			
87	8	1	87	-32768			
88	1333	1435	0	1	2	4000	3
89	-32768	-32768	-32768	-32768	-32768	-32768	-32768
90	4	7	23	-32768			
91	3	-32768	-32	-32768			
92	7	3	57	-32768			
93	274	220	0	65	2	5000	4
94	63	42	-32768	-32768	-32768	-32768	-32768
95	3	3	21	-32768			
96	5	17	32	-32768			
97	1	7	-37	-32768			
98	1	2	430	-32768			
99	1443	1445	0	68	2	1700	4
100	63	42	-32768	-32768	-32768	-32768	-32768
101	7	3	21	-32768			
102	5	17	32	-32768			
103	1	7	-37	-32768			
104	1	2	430	-32768			
105	895	306	0	1424	2	1000	4
106	63	42	-32768	-32768	-32768	-32768	-32768
107	3	3	21	-32768			
108	5	17	32	-32768			
109	1	7	-37	-32768			
110	1	2	430	-32768			
111	1775	1244	0	25	2	299	4
112	63	42	-32768	-32768	-32768	-32768	-32768
113	3	3	21	-32768			
114	5	17	32	-32768			
115	1	7	-37	-32768			
116	1	2	430	-32768			
117	617	484	0	67	3	17	0

CFAS DATA ELEMENT OBSREP

118	4	8	2	7	6	4	0
119	5	-32768	-32768	-32768	-32768	-32768	-32768
120	1374	248	0	26	3	9	0
121	4	8	2	7	6	4	4
122	5	41	-32768	-32768	-32768	-32768	-32768
123	1932	1315	0	198	3	9	0
124	4	8	-32768	7	6	4	4
125	5	41	-32768	-32768	-32768	-32768	-32768
126	1435	519	0	118	3	4	0
127	4	8	2	7	6	4	4
128	5	44	-32768	-32768	-32768	-32768	-32768
129	184	607	0	56	3	8	0
130	4	8	2	7	6	4	4
131	5	43	-32768	-32768	-32768	-32768	-32768
132	1476	1311	0	1416	-3	25	0
133	-32768	4	-32768	8	-32768	-32768	8
134	25	-32768	-32768	-32768	-32768	-32768	-32768
135	1493	385	0	7	3	9	0
136	-32768	4	-32768	8	-32768	-32768	8
137	25	42	-32768	-32768	-32768	-32768	-32768
138	1618	1249	0	1407	3	4	0
139	-32768	4	-32768	8	-32768	-32768	8
140	25	43	-32768	-32768	-32768	-32768	-32768
141	192	1668	0	29	3	4	0
142	-32768	4	-32768	8	-32768	-32768	8
143	-32768	43	-32768	-32768	-32768	-32768	-32768
144	125	947	0	49	3	20	0
145	8	-32768	9	-32768	0	0	3
146	35	-32768	-32768	-32768	-32768	-32768	-32768
147	476	276	0	1432	3	20	0
148	8	5	2	3	6	-32768	3
149	5	-32768	-32768	-32768	-32768	-32768	-32768
150	1602	1648	0	80	3	9	0
151	8	-32768	9	-32768	0	0	3
152	35	44	-32768	-32768	-32768	-32768	-32768
153	1839	770	0	1412	3	3	0
154	8	-32768	9	-32768	0	0	3
155	35	46	-32768	-32768	-32768	-32768	-32768
156	1543	766	0	102	3	3	0
157	8	-32768	9	-32768	0	0	3
158	46	-32768	-32768	-32768	-32768	-32768	-32768
159	1575	650	0	15	3	25	0
160	8	8	0	7	9	1	5
161	55	-32768	-32768	-32768	-32768	-32768	-32768
162	430	286	0	115	3	9	0
163	8	8	0	7	9	1	5
164	55	42	-32768	-32768	-32768	-32768	-32768
165	530	1193	0	37	3	4	0
166	8	8	0	7	9	1	5
167	55	43	-32768	-32768	-32768	-32768	-32768
168	570	332	0	100	3	21	0
169	1	-32768	2	7	1	2	0
170	5	-32768	-32768	-32768	-32768	-32768	-32768
171	287	1838	0	41	3	9	0
172	1	-32768	2	7	1	2	4
173	5	41	-32768	-32768	-32768	-32768	-32768
174	138	619	0	15	3	4	0
175	1	-32768	2	7	1	2	4
176	5	44	-32768	-32768	-32768	-32768	-32768

CFAC DATA ELEMENT OBSREP

177	1246	515	0	68	3	21	0
178	2	4	3	-32768	3	4	8
179	25	-32768	-32768	-32768	-32768	-32768	-32768
180	893	103	0	1412	3	9	0
181	2	4	3	-32768	3	4	8
182	25	41	-32768	-32768	-32768	-32768	-32768
183	1267	1870	0	81	3	4	0
184	2	4	3	-32768	3	4	8
185	25	46	-32768	-32768	-32768	-32768	-32768
186	2	1102	0	19	3	4	0
187	2	4	3	-32768	3	4	8
188	46	-32768	-32768	-32768	-32768	-32768	-32768
189	376	1352	0	44	3	12	0
190	3	8	9	8	5	6	3
191	35	-32768	-32768	-32768	-32768	-32768	-32768
192	1217	143	0	69	3	9	0
193	3	8	9	8	5	6	4
194	35	46	-32768	-32768	-32768	-32768	-32768
195	1371	651	0	30	3	9	0
196	3	8	7	8	5	5	4
197	35	46	-32768	-32768	-32768	-32768	-32768
198	122	836	0	36	3	4	0
199	3	8	9	8	5	8	4
200	35	45	-32768	-32768	-32768	-32768	-32768
201	974	438	0	77	3	21	0
202	4	-32768	2	-32768	7	8	5
203	55	-32768	-32768	-32768	-32768	-32768	-32768
204	1401	853	0	33	3	90	0
205	4	-32768	2	-32768	7	8	5
206	55	43	-32768	-32768	-32768	-32768	-32768
207	1202	46	0	25	3	4	0
208	4	-32768	2	-32768	7	8	5
209	55	44	-32768	-32768	-32768	-32768	-32768
210	50	1298	0	129	3	21	0
211	5	4	3	7	9	1	6
212	65	-32768	-32768	-32768	-32768	-32768	-32768
213	116	1771	0	122	3	9	0
214	5	4	3	7	9	1	6
215	65	42	-32768	-32768	-32768	-32768	-32768
216	530	1631	0	95	3	4	0
217	5	4	3	7	9	1	6
218	65	45	-32768	-32768	-32768	-32768	-32768
219	1391	126	0	50	3	9	0
220	6	8	9	-32768	2	3	7
221	75	-32768	-32768	-32768	-32768	-32768	-32768
222	1923	308	0	1423	3	9	0
223	6	8	9	-32768	2	3	7
224	75	48	-32768	-32768	-32768	-32768	-32768
225	408	1450	0	11	3	4	0
226	6	8	9	-32768	2	3	7
227	75	49	-32768	-32768	-32768	-32768	-32768
228	244	353	0	1434	3	21	0
229	7	-32768	2	8	4	5	8
230	35	-32768	-32768	-32768	-32768	-32768	-32768
231	104	292	0	4	3	9	0
232	7	-32768	2	8	4	5	8
233	85	42	-32768	-32768	-32768	-32768	-32768
234	1423	1944	0	124	3	4	0
235	7	-32768	2	8	4	5	8

CFAS DATA ELEMENT OBSREP

236	85	46	-32768	-32768	-32768	-32768	-32768
237	276	417	0	16	3	21	0
238	8	4	3	8	6	7	9
239	95	-32768	-32768	-32768	-32768	-32768	-32768
240	626	1551	0	49	-3	9	0
241	8	4	3	8	6	7	9
242	95	42	-32768	-32768	-32768	-32768	-32768
243	905	1573	0	46	-3	4	0
244	8	4	3	8	6	7	9
245	95	43	-32768	-32768	-32768	-32768	-32768
246	1501	51	0	85	-3	32	0
247	4	4	9	5	0	3	9
248	91	-32768	-32768	-32768	-32768	-32768	-32768
249	1773	1549	0	1395	3	48	0
250	7	3	6	4	2	8	2
251	21	-32768	-32768	-32768	-32768	-32768	-32768
252	914	805	0	20	3	15	0
253	8	8	7	3	7	-32768	5
254	50	-32768	-32768	-32768	-32768	-32768	-32768
255	682	825	0	70	-3	8	0
256	6	4	3	5	8	-32768	6
257	62	-32768	-32768	-32768	-32768	-32768	-32768
258	1266	1939	0	1437	-3	4	0
259	8	6	0	4	7	-32768	7
260	75	-32768	-32768	-32768	-32768	-32768	-32768
261	731	1543	0	25	3	15	0
262	8	8	7	3	7	-32768	5
263	50	44	-32768	-32768	-32768	-32768	-32768
264	945	766	0	72	4		
265	0	10040	2839	158			
266	12	10000	2829	150			
267	-32768	8730	2718	75			
268	144	8500	2710	114			
269	-32768	7750	2675	51			
270	-32768	7620	2699	183			
271	-32768	7450	2705	173			
272	297	7000	2679	170			
273	-32768	6500	2647	166			
274	-32768	5900	2605	75			
275	-32768	5700	2594	118			
276	554	5000	2521	123			
277	-32768	4400	2442	-32768			
278	714	4000	2389	-32768			
279	910	3000	2270	-32768			
280	1030	2500	2221	-32768			
281	1175	2000	2230	-32768			
282	1362	1500	2211	-32768			
283	1621	-1000	2151	-32768			
284	72	134	0	79	4		
285	0	-32768	2839	158			
286	12	10000	2829	150			
287	-32768	8730	2718	75			
288	144	8500	2710	114			
289	-32768	7750	2675	51			
290	-32768	7620	2699	183			
291	-32768	7450	2705	173			
292	297	7000	2679	170			
293	-32768	6500	2647	166			
294	-32768	5980	2605	75			

CFAS DATA ELEMENT OBSREP

295	-32768	5760	2594	118	
296	554	5000	2521	123	
297	-32768	4400	2442	-32768	
298	714	4000	2339	-32768	
299	910	3000	2270	-32768	
300	1030	2500	2221	-32768	
301	1175	2000	2230	-32768	
302	1362	1500	2211	-32768	
303	1621	-1000	2151	-32768	
304	1910	217	0	1423	4
305	0	10040	-32768	158	
306	12	10000	2829	150	
307	-32768	8730	2718	75	
308	144	8500	2710	114	
309	-32768	7760	2675	51	
310	-32768	7620	2699	183	
311	-32768	7460	2705	173	
312	297	7000	2679	170	
313	-32768	6500	2647	166	
314	-32768	5980	2605	75	
315	-32768	5760	2594	118	
316	554	5000	2521	123	
317	-32768	4400	2442	-32768	
318	714	4000	2339	-32768	
319	910	3000	2270	-32768	
320	1030	2500	2221	-32768	
321	1175	2000	2230	-32768	
322	1362	1500	2211	-32768	
323	1621	-1000	2151	-32768	
324	1001	1433	0	1424	4
325	0	-32768	-32768	158	
326	12	10000	2829	150	
327	-32768	8730	2718	75	
328	144	8500	2710	114	
329	-32768	7760	2675	51	
330	-32768	7620	2699	183	
331	-32768	7460	2705	173	
332	297	7000	2679	170	
333	-32768	6500	2647	166	
334	-32768	5980	2605	75	
335	-32768	5760	2594	118	
336	554	5000	2521	123	
337	-32768	4400	2442	-32768	
338	714	4000	2339	-32768	
339	910	3000	2270	-32768	
340	1030	2500	2221	-32768	
341	1175	2000	2230	-32768	
342	1362	1500	2211	-32768	
343	1621	-1000	2151	-32768	
344	1123	46	0	9	4
345	-32768	-32768	2839	158	
346	-32768	10000	2829	150	
347	-32768	8730	2718	75	
348	-32768	8500	2710	114	
349	-32768	7760	2675	51	
350	-32768	7620	2699	183	
351	-32768	7460	2705	173	
352	-32768	7000	2679	170	
353	-32768	6500	2647	166	

CFAS DATA ELEMENT OBSREP

354	-32768	5980	2605	75	
355	-32768	5760	2594	118	
356	-32768	5000	2521	123	
357	-32768	4400	2442	-32768	
358	-32768	4000	2389	-32768	
359	-32768	3000	2270	-32768	
360	-32768	2500	2221	-32768	
361	-32768	2000	2230	-32768	
362	-32768	1500	2211	-32768	
363	-32768	-1000	2151	-32768	
364	531	1318	0	1439	4
365	-32768	-32768	-32768	158	
366	-32768	10000	2829	150	
367	-32768	8730	2718	75	
368	-32768	8500	2710	114	
369	-32768	7760	2675	51	
370	-32768	7620	2699	193	
371	-32768	7460	2705	173	
372	-32768	7000	2679	170	
373	-32768	6500	2647	166	
374	-32768	5980	2605	75	
375	-32768	5760	2594	118	
376	-32768	5000	2521	123	
377	-32768	4400	2442	-32768	
378	-32768	4000	2389	-32768	
379	-32768	3000	2270	-32768	
380	-32768	2500	2221	-32768	
381	-32768	2000	2230	-32768	
382	-32768	1500	2211	-32768	
383	-32768	-1000	2151	-32768	
384	1141	1792	0	91	4
385	-32768	-32768	2839	158	
386	-32768	10000	2829	150	
387	-32768	8730	2718	75	
388	-32768	8500	2710	114	
389	-32768	7760	2675	51	
390	-32768	7620	2699	183	
391	-32768	7460	2705	173	
392	297	7000	2679	170	
393	-32768	6500	2647	166	
394	-32768	5980	2605	75	
395	-32768	5760	2594	118	
396	554	5000	2521	123	
397	-32768	4400	2442	-32768	
398	714	4000	2389	-32768	
399	910	3000	2270	-32768	
400	1030	2500	2221	-32768	
401	1175	2000	2230	-32768	
402	1362	1500	2211	-32768	
403	1621	-1000	2151	-32768	
404	662	256	0	1427	4
405	-32768	-32768	-32768	158	
406	-32768	10000	2829	150	
407	-32768	8730	2718	75	
408	-32768	8500	2710	114	
409	-32768	7760	2675	51	
410	-32768	7620	2699	183	
411	-32768	7460	2705	173	
412	297	7000	2679	170	

CFAS DATA ELEMENT OBSREP

413	-32768	6500	2647	166	
414	-32768	5930	2605	75	
415	-32768	5760	2594	118	
416	554	5000	2521	123	
417	,-32768	4400	2442	-32768	
418	714	4000	2389	-32768	
419	910	3000	2270	-32768	
420	1030	2500	2221	-32768	
421	1175	2000	2230	-32768	
422	1362	1500	2211	-32768	
423	1621	-1000	2151	-32768	
424	347	1576	0	45	4
425	0	10040	2839	158	
426	12	10000	-32768	150	
427	-32768	8730	-32768	75	
428	144	8500	-32768	114	
429	-32768	7760	2675	51	
430	-32768	7620	-32768	183	
431	-32768	7460	-32768	173	
432	297	7000	-32768	170	
433	-32768	6500	-32768	166	
434	-32768	5930	2605	75	
435	-32768	5760	2594	118	
436	554	5000	2521	123	
437	-32768	4400	2442	-32768	
438	714	4000	2389	-32768	
439	910	3000	2270	-32768	
440	1030	2500	2221	-32768	
441	1175	2000	2230	-32768	
442	1362	1500	2211	-32768	
443	1621	-1000	2151	-32768	
444	1334	1333	0	105	44
445	0	10040	2839	158	
446	12	10000	2829	150	
447	-32768	8730	2718	75	
448	144	-32768	2710	114	
449	-32768	7760	2675	51	
450	-32768	7620	2699	183	
451	-32768	-32768	2705	173	
452	297	7000	2679	170	
453	-32768	6500	2647	166	
454	-32768	5930	2605	75	
455	-32768	5760	2594	118	
456	554	-32768	2521	123	
457	-32768	4400	2442	-32768	
458	714	4000	2389	-32768	
459	910	3000	2270	-32768	
460	1030	2500	2221	-32768	
461	1175	2000	2230	-32768	
462	1362	1500	2211	-32768	
463	1621	-1000	2151	-32768	

BEST AVAILABLE COPY

```
CFAS DATA ELEMENT TR0010

CLOUD-FOG*CFAS.TR0010
                1            0         1410
                2            0         1420
          &ADD,P CFAS.OBSREP
         .      3            0          160
              160         1425            4            1            2            3            4
             20.0         20.0         80.0        100.0         50.0        120.0        150.0
                4            0          170
              170         1425            4            1            2            3            4
             20.0         20.0         80.0        100.0         50.0        120.0        150.0
             51.0         51.0         99.0         99.0
```

I-103

CFAS RUNSTREAM ELEMENT STORE

```
CLOUD-FOG*CFAS.STORE
     1       @ASG,T DISK0,F/1/
     2       @ASG,T DISK1,F/1/
     3       @ASG,T DISK2,F/1/
     4       @ASG,T DISK3,F/1/
     5       @ASG,T DISK4,F/1/
     6       @USE F0,DISK0
     7       @USE F1,DISK1
     8       @USE F2,DISK2
     9       @USE F3,DISK3
    10       @USE F4,DISK4
```

CFAS RUNSTREAM ELEMENT TR001C

```
CLOUD-FOG*CFAS.TR001C
        @ADD CFAS.STORE
        @MAP,NI
         IN CFAS.CFMAIN
         LIB CFAS.
        @XQT
        @ADD,P CFAS.TR001D
```

APPENDIX II

NUMERICAL CODES FOR CFAS OF AIRWAYS DATA ELEMENTS

TABLE II-1

CFAS CODE 1

NUMERICAL CODIFICATION FOR CFAS OF

AIRWAYS WEATHER AND OBSTRUCTION TO VISION SYMBOLS

AIRWAYS CODE	DESCRIPTION	NEAREST WMO CODE 4677	CFAS CODE
K	Smoke	04	04
H	Haze	05	05
D	Dust	06 - 07	07
GF	Ground Fog	11 - 12	12
BD	Blowing Dust	30 - 35	30
BN	Blowing Sand	30 - 35	35
BS	Blowing Snow	38 - 39	39
BY	Blowing Spray	None	None
F	Fog	41 - 49	45
IF	Ice Fog	41 - 49	47
L- -	Drizzle, very light	50	50
L-	Drizzle, light	51	51
L	Drizzle, moderate	52 - 53	53
L+	Drizzle, heavy	54 - 55	55
ZL--	Freezing Drizzle, very light	56	56
ZL-	Freezing Drizzle, light	56	56
ZL	Freezing Drizzle, moderate	57	57
ZL+	Freezing Drizzle, heavy	57	57
R--	Rain, very light	60	60
R-	Rain, light	61	61
R	Rain, moderate	62 - 63	63
R+	Rain, heavy	64 - 65	65
ZR--	Freezing Rain, very light	66	66
ZR-	Freezing Rain, light	66	66
ZR	Freezing Rain, moderate	67	67
ZR+	Freezing Rain, heavy	67	67

TABLE II-1 (Continued)

CFAS CODE 1

NUMERICAL CODIFICATION FOR CFAS OF

AIRWAYS WEATHER AND OBSTRUCTION TO VISION SYMBOLS

AIRWAYS CODE	DESCRIPTION	NEAREST WMO CODE 4677	CFAS CODE
S--	Snow, very light	70	70
S-	Snow, light	71	71
S	Snow, moderate	72 - 73	72 - 73
S+	Snow, heavy	74 - 75	74 - 75
SG--	Snow Grains, very light	77	77
SG-	Snow Grains, light	77	177
SG	Snow Grains, moderate	77	277
SG+	Snow Grains, heavy	77	377
SP--	Snow Pellets, very light	79	79
SP-	Snow Pellets, light	79	179
SP	Snow Pellets, moderate	79	279
SP+	Snow Pellets, heavy	79	379
IC	Ice Crystals	76 or 78	78
IP--	Ice Pellets, very light	79	79
IP-	Ice Pellets, light	79	179
IP	Ice Pellets, moderate	79	279
IP+	Ice Pellets, heavy	79	379
IPW--	Ice Pellet Showers, very light	87	87
IPW-	Ice Pellet Showers, light	87	187
IPW	Ice Pellet Showers, moderate	88	88
IPW+	Ice Pellet Showers, heavy	88	188
RW--	Rain Showers, very light	80	80
RW-	Rain Showers, light	80	180
RW	Rain Showers, moderate	81	81
RW+	Rain Showers, heavy	82	82

TABLE II-1 (Continued)

CFAS CODE 1

NUMERICAL CODIFICATION FOR CFAS OF

AIRWAYS WEATHER AND OBSTRUCTION TO VISION SYMBOLS

AIRWAYS CODE	DESCRIPTION	NEAREST WMO CODE 4677	CFAS CODE
SW--	Snow Showers, very light	85	85
SW-	Snow Showers, light	85	185
SW	Snow Showers, moderate	86	86
SW+	Snow Showers, heavy	86	186
A--	Hail, very light	89	89
A-	Hail, light	89	189
A	Hail, moderate	90	90
A+	Hail, heavy	90	190
T	Thunderstorm, light or moderate	95 - 96	96
T+	Thunderstorm, severe	99	99

TABLE II-2

CFAS CODE 2

NUMERICAL CODIFICATION FOR CFAS OF AIRWAYS SKY COVER SYMBOLS

AIRWAYS CODE	DESCRIPTION	NEAREST WMO CODE 2700	CFAS CODE
- X	Partly obscured sky	None	9
X	Totally obscured sky	9	9
○	Clear sky	0	0
⦶	Scattered (0.1-0.5 sky cover)	1 - 4	3
⦷	Broken (0.6-0.9 sky cover)	5 - 7	6
⊕	Overcast (1.0 sky cover)	8	8

For a minus sign (-) preceding ⦶ , ⦷ or ⊕, set variable name ITHN(I) = 1.